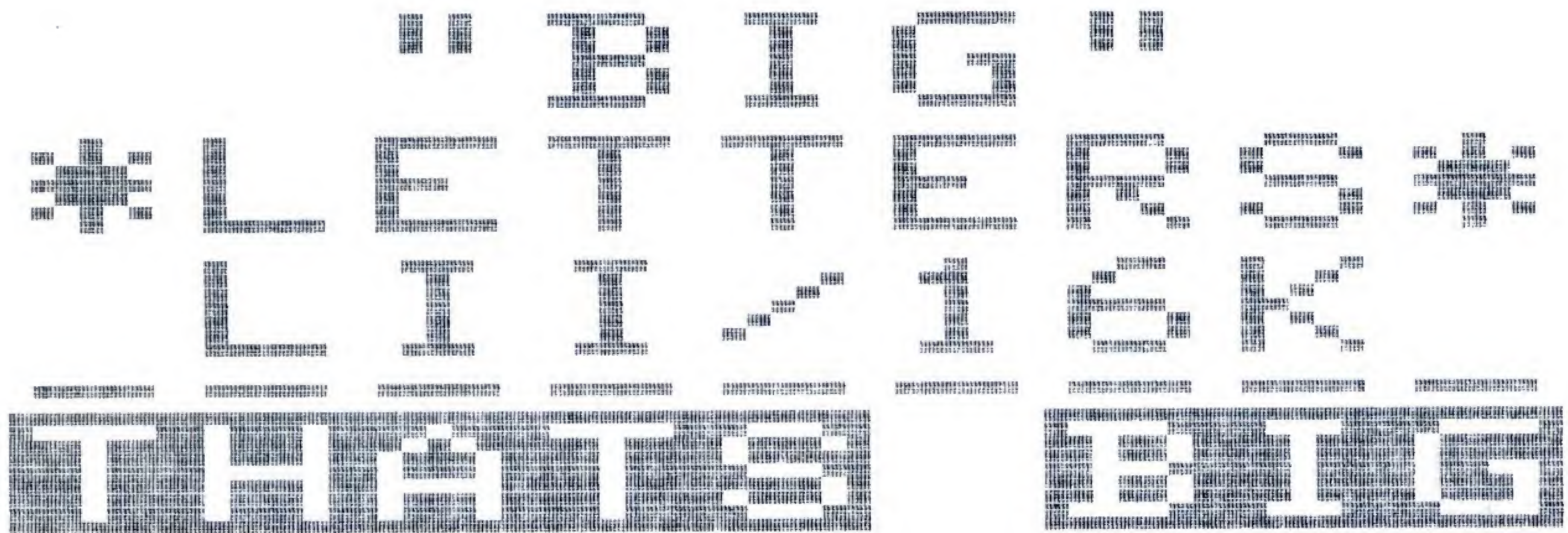


Issue 21, August 1981



MESSAGES YOU CAN REALLY SEE!!

Also in this issue:

HARDWARE:

Add Joysticks and Input/Output Ports to your
'80 — Part 2

PROGRAMMING:

Better Basic Programming — Part 4

SOFTWARE:

Birthday — Level 1
Binary Converter
Cosmic Wars

Guzinta — Level 1
Linear Regression
Util 1, 2, 3 & 4

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034. AUSTRALIA. TELEPHONE (08) 211 7244. PRICE: AUS. \$2.50, N.Z. \$4.00, U.K. £1.50

***** ABOUT MICRO-80 *****

EDITOR:	IAN VAGG
SOFTWARE EDITOR:	CHARLIE BARTLETT
HARDWARE EDITOR:	EDWIN PAAY
U.K. CORRESPONDENT:	TONY EDWARDS

MICRO-80 is an international magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80/Video Genie. It is available at the following prices (all prices shown in Aus.\$ except for U.K. prices which are in pounds Sterling).

12 months subscription	Aus.	\$24.00
	NZ.	\$36.00 (Airmail)
	Hong Kong	\$46.00 (Airmail)
	U.K.	£16.00
Single Copy	Aus.	\$2.50
	N.Z.	\$3.50 (Airmail)
	Hong Kong	\$4.25 (Airmail)
	U.K.	£1.50
Months programs on cassette	Aus.	\$3.50
	N.Z.	\$4.00 (Airmail)
	Hong Kong	\$4.50 (Airmail)
(at present available from Australia only)	U.K.	\$4.75 (Airmail)
12 months subscription to magazine and cassette	Aus.	\$60.00
	N.Z.	\$78.00 (Airmail)
	Hong Kong	\$88.00 (Airmail)
	U.K.	£41.00 (Airmail)

Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80, System 80 or Video Genie and their peripherals. MICRO-80 is in no way connected with either the Tandy or Dick Smith organisations.

**** WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS ****

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-80 or System 80 to earn some extra income is included in every issue.

**** CONTENT ****

Each month we publish at least one applications program in Level I BASIC, one in Level II BASIC and one in DISK BASIC (or disk compatible Level II). We also publish Utility programs in Level II BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

**** COPYRIGHT ****

All the material published in this magazine is under copyright. That means that you must not copy it, except for your own use. This applies to photocopying the magazine itself or making copies of programs on tape or disk.

**** LIABILITY ****

The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

***** CONTENTS *****

	Page
EDITORIAL	2
PEEKing (FROM OUR U.K. CORRESPONDENT)	2
USING LEVEL 2 PROGRAMS FROM DISK	3
MARKET PLACE	4
BETTER BASIC PROGRAMMING — Part 4	5
REVERSING THE ROLE OF PRINT AND LPRINT STATEMENTS	6
INPUT/OUTPUT (Letters to the Editor)	7
JOYSTICKS AND INPUT/OUTPUT PORTS FOR YOUR '80	7
 <u>SOFTWARE SECTION</u>	
GUZINTA.....L1/4K	11
BIRTHDAYL1/4K	13
BINARY CONVERTERL2/4K	14
LINEAR REGRESSIONL2/4K	15
BIG LETTERSL2/16K	21
COSMIC WARS.....DISK/32K	27
UTIL 1: UTIL 2: UTIL 3: UTIL 4L2/m.l.	28
 NEXT MONTH'S ISSUE	
CASSETTE EDITION INDEX	36
ORDER FORM	36

MICRO-80 is registered by Australia Post - Publication SQB 2207 Category B

AUSTRALIAN OFFICE AND EDITOR:

MICRO-80, P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA, 5034. TEL. (08) 211 7244

U.K. SUBSCRIPTION DEPT:

24 WOODHILL PARK, PEMBURY, TUNBRIDGE WELLS, KENT TN2 4NW.

Printed by:

Shovel & Bull Printers, 312A Unley Road, HYDE PARK, S.A. 5061.

Published in Australia by MICRO-80, 433 Morphett Street, ADELAIDE.

★ ★ ★ FREE SOFTWARE OFFER ★ ★ ★

EVERY NEW SUBSCRIBER TO MICRO-80 WILL RECEIVE A FREE CASSETTE
CONTAINING THREE LEVEL I AND THREE LEVEL II PROGRAMS PLUS COMPREHENSIVE
DOCUMENTATION ... THE RETAIL VALUE OF THE SOFTWARE WOULD EXCEED
THE COST OF THE SUBSCRIPTION!!!

***** EDITORIAL *****

There are some interesting changes afoot in the world of microcomputing. Whilst microcomputers as we know them are continuing on with the TRS-80 Model III, Apple III (or IV), Commodore etc., there is a whole new breed of microcomputer just about to appear in Australia and the U.K. These are the new generation colour computers. Spawned by the popular video games, these colour computers generally have high resolution graphics and, of course, colour capability.

Human beings are reactionary creatures and, I suspect, many of us will look down our noses at these new-fangled "toys" and consider them beneath us. There is an interesting parallel here with the automotive industry. Consider what happened to the British open sports cars of the 50's and 60's. They were the epitome of performance vehicles in their day but the majority of people bought closed, saloon cars. Manufacturers could afford to invest large sums to develop the performance of these machines whilst sports cars, with their small market base, remained relatively stagnant in design. Today, a good G.T. saloon will outperform most of the few remaining sports cars which, one by one, are disappearing.

We may see the same situation occur with microcomputers. If they really do sell in the vast numbers that their manufacturers obviously expect, then the money will be available to develop them and their performance will ultimately approach and then exceed that of the more "serious" computers. Personally, I do not believe that this will be any bad thing. Most programs, be they games or business applications, are improved by the use of colour and high resolution graphics. Imagine an inventory control program which shows out-of-stock items in red, for example, or a debtors' ledger which shows current account in green, thirty day accounts in orange and 60 day or greater accounts in red. Such features would be of considerable practical value.

The first three contenders to appear in Australia will be the Atari, the Commodore VIC-20 and the TRS-80 Colour. Starting prices, excluding the colour monitor, range from \$399 to \$599, so they will be quite cheap. The TRS-80 Colour, in particular, is capable of considerable expansion, including the addition of disk drives. It is interesting to note that software houses in the U.S.A. are releasing programs for the TRS-80 Colour (which is not compatible with other TRS-80's - it uses a 6809 microprocessor chip, for one thing). Although only a trickle as yet, the signs are that there will eventually be a flood of programs for these machines. We do not intend to ignore the trend. Shortly, we will publish a review of the TRS-80 Colour written by a reader who imported his model directly from the U.S.A. some months ago. As the number of colour machines grow, we will also publish suitable software for them.

- 0000000000 -

***** PEEKing (U.K.) *****

From our U.K. correspondent Tony Edwards

Our non-U.K. readers may be interested to hear a little about the set-up of the '80 in the United Kingdom so, in this early article, I am going to summarize the position. The TRS-80 is widely available from Tandy via its numerous 'Radio Shack' outlets and through 'Tandy Computer Centres' which are now appearing in many areas. Models I and II are available and widely used, and Model III is now becoming available. The approximate costs of these units are £390, £440 and £700 respectively. In addition, a Model III (Level I) at £500 is also advertised. These prices, of course, vary considerably with the amount of memory included, but those quoted are for the smallest memory options. With the new Model III in use, rumours are circulating that they are not compatible with Model I programmes. I understand that utility firmware addresses differ between the two machines and that this, together with the 258 bytes reduction in memory on the Model III, may cause some trouble. Model I discs can mostly be converted to Model III using a Convert utility, but Tandy admit that not all its software can be so converted, and they provide a list - but how about non-Tandy software? Tandy also provide printers with the 'Lineprinter VI' (132 column, bidirectional) selling at about £600, and a new 'Tandy Printer VII' (pinfeed plain paper) now reaching the market at about £200.

The machines sold in Australia as 'System 80' are also available in the U.K. but are known as the 'Video Genie' in this country. They are imported by Lowe Electronics and marketed through a large number of independent dealer outlets. This leads to a variance in pricing but as a guide, the basic Genie costs some £300 and the necessary monitor about £80. Early Genies were similar to the early System 80's and suffered the same faults, but those now being marketed have all four 'arrow' keys, a 'clear' key and a UV play level meter on the internal cassette. Upgrade kits are available for older machines at quite reasonable costs. My 'Genie' is the latest model with the meter and I can report that I have no difficulties at all loading tapes recorded on a number of other machines. The latest change from Lowe Electronics is a colour Video Genie together with a colour update kit at about £35 for the kit. This colour Genie has had good reports, but I have not yet had a chance to use one myself.

A large number of peripherals is available for the '80, some from Tandy but many from a host of small firms supplying '80 compatible hardware. "Stringy Floppies" are, for instance, available at £180 for both types of machine. U.K. users are also very well provided for in respect of software, but I will leave that until some future issue.

(This colour version of the Video Genie is not yet available in Australia. We would welcome information concerning this machine from any of our U.K. readers who own or use one - Ed.)

***** USING LEVEL 2 PROGRAMS FROM DISK *****

Disks are great for speed and reliability but, every once in a while, you come across a Level 2 program which will not RUN under Disk BASIC. Usually, this is because there are machine language routines accessed by a USR call. Recently, with the advent of sound routines in many programs, the situation has become worse. Often, the machine language sound routines are POKEd into line 0 of the BASIC program which starts at 42E9, squarely in the DOS RAM area and therefore unusable. So, how do you RUN such programs in Disk BASIC without extensive modifications? The answer is, you don't, you use a trick to LOAD them from Disk and then RUN them in Level 2 BASIC.

The idea is to save onto disk a block of memory containing the Level 2 system RAM and the Level 2 BASIC program, just before it is RUN, with a small block move routine appended and the filespec extension CMD. When you wish to RUN the program, you simply type in its filespec, the file is loaded into high memory, the block move routine moves it into its correct position, overwriting system RAM and DOS in the process then enters Level 2 BASIC. At this point, the computer thinks it is a Level 2 machine with the original Level 2 program all set to RUN and so it is. Type RUN press ENTER/NEWLINE and away you go.

There are several different ways to achieve the above, depending on the utility programs you have available. We chose to use BMON, published in issues 3, 4 and 5 of MICRO-80 or available on cassette from MICRO-80 PRODUCTS and LMOFFSET, a program which comes with NEWDOS + or NEWDOS 80. The following is the step-by-step procedure -

1. Set up the computer complete with disk system. Press BREAK/RESET or type in BASIC 2 from DOS.
2. Answer Ready?/Memory Size? with the value your Level 2 program will eventually require. If it is not necessary to protect memory size for your Level 2 program, simply press ENTER/NEWLINE. Warning: There is a trap here. You are going to use BMON in a moment. Do not protect BMON; it is not necessary in this instance and whatever Memory Size you set now will be reproduced later when you load your program from disk. If you protect BMON you may encounter an OM error when RUNning your Level 2 program.
3. Load the version of BMON into your machine which will allow space for the Level 2 program. Usually the 16K version will do but, if not, use the 32K or 48K version.
4. Use the L option in BMON to load in the Level 2 BASIC program from cassette.
5. When the program has finished loading, the Start and End addresses will be shown. The Start address should be 42E9 Hex, note down the End address. In our example, illustrated below, the end address is 6086 Hex.
6. Use the Copy Memory function in BMON to punch out a SYSTEM tape starting at 4000 Hex, ending at the end of the BASIC program and having an Entry point of 06CC Hex. This tape may have any convenient name you choose as it will be discarded later. The purpose of punching out this tape is to enable you to recreate the exact situation that existed in the normal Level 2 machine at the instant the program was ready to RUN. The entry address 06CC Hex is one of the available entry points into the ROM to pass control back to BASIC (according to Eddy Paay, the best address to use - Ed.)
7. Make sure that you have a SYSTEM disk in drive 0, that LMOFFSET is available on one of the mounted disks and that you have sufficient space on one of the non write-protected disks to write the program plus 760 bytes. Reboot DOS, TYPE IN LMOFFSET, press ENTER/NEWLINE and then follow the screen format shown below.
8. APPARAT LOAD MODULE OFFSET PROGRAM, VERSION 1.1 ***
 SOURCE FROM DISK OR TAPE? REPLY "D" OR "T"?T
 MODULE LOADS TO 4000-6086
 MODULE LOAD OVERLAPS DOS RAM (4000-51FF)
 MODULE LOAD WILL OVERLAP "CMD" PROGRAM AREA (5200-6FFF)
 ENTRY POINT = 06CC
 NEW LOAD BASE ADDRESS (HEX)? 7000
 SHALL APPENDAGE BE SUPPRESSED (Y OR N)?N
 MODULE LOADS TO 7000-9015
 ENTRY POINT = 9087
 NEW LOAD BASE ADDRESS (HEX)? (ENTER IS PRESSED HERE)
 INTERRUPTS TO BE DISABLED (Y OR N)?Y
 DESTINATION FILESPE? MYPROG/CMD
 MODULE WRITE COMPLETED
 "ENTER" TO RESTART PROGRAM ANEW? (REBOOT HERE)
9. The exercise is now completed. Whenever you want to use the program, mount the disk, type in MYPROG, press ENTER/NEWLINE answer READY with RUN, just as if you had loaded it into a Level 2 machine from tape.

***** MARKET PLACE *****

Market Place is available to any reader who has hardware to dispose of. An entry costs nothing - you pay MICRO-80 \$5.00 or 5% commission, whichever is the greater - up to a maximum of \$30, after the goods are sold. The commission is calculated on your advertised price.

SYSTEM-80 Level 2 16K with input level indicator, right arrow and clear keys, I/P changeover switch plus Emerson cassette recorder for external input, plus 12" B & W T.V. monitor with video I/P conversion, plus 13 months of MICRO-80 issues and cassettes, plus 10 Scotch computer cassettes \$700.00
the lot

Mr. B. Cameron,
15 George Street,
Pennant Hills, NSW. 2120.

SYSTEM-80 LEVEL II 16K MEMORY, GAMES, CASSETTES, BOOKS AND MANUALS, brand new \$680.00

Mr. M. Gosbee. Ph. (02) BB-446B

AXION SERIAL THERMAL PRINTER with B rolls of paper and TRS-232 interface \$400.00

Mr. T. Domigan,
P.O. Box 390986,
Winnellie, NT. 5789.

Radio Shack QUICK PRINTER: 150 LPM on 4.75" wide aluminised paper. Software selectable 80, 40 or 20 characters per line. Upper-case with full descenders. Automatic underlining. Seldom used and in excellent working order. Catalogue price \$699. Will sell for: \$400.00

K. Cook,
P.O. Box 931,
Gove, NT. 5797.

ITOH DOT MATRIX PRINTER. 80 Columns wide with cable for TRS-80, Good working condition \$750.00

Mr. Pat Browne,
90 Muriel Drive,
Pooraka, SA. 5095.
Ph. 258-1555 Bus. 262-4020 A.H.

SYSTEM-80 L2/16K with full documentation, approx. 40 blank cassettes (data), books, magazines, and programs (most games) \$720.00
including EDTASM PLUS & Adventure NO. 9 o.n.o.

Mr. R. Glucz,
1/1707 Dandenong Rd.
Oakleigh, Vic. 3166.
Ph. (03) 543-3302 A.H.

TRS-80 MODEL 1 Level II 4K excellent condition, Catalogue price \$879. Sell with manuals and programs \$750.00

James Grigg.
Ph. (052) 7B-5056.

TRS-80 LEVEL II 16K, green screen; LNW expansion in attractive box complete functions except dual cassette control, PERTEC FD200 disk drives - 2 units each containing 2 40 track drives, cables plus CRT-41 cassette \$3,000.00

Mr. B. Clark,
P.O. box 46,
Dee Why, NSW. 2099.
Ph. (02) 9B1-1114

SYSTEM-80 L2/16K Mark II model with input level indicator, right arrow & clear keys, switch for second cassette recorder, jack for external sound amplifier, with full documentation, dust cover, 12" B&W TV monitor, \$150 in programs, Sony TC-11S cassette recorder, MICRO-80 & electronics magazines baluns etc. & much more. Excellent condition. Catalogue price \$1,149. Will sell for \$875.00
the lot o.n.o.

Mr. M. Bellamy.
Ph. (03) 52B-6143.

KTM-2 KEYBOARD UNIT in good order \$235.00

Mr. A.R. Hall,
c/o P.O. Southbrook, Qld. 4352.

TANDY MODEL III complete with 4B K RAM, line printer (7) and System Desk \$1,850.00

Mike Mooney,
(02) 76-B112 A.H.
(02) B2-1943 Bus.

SYSTEM 80 16K Computer with video screen, Audio amplifier and 15 issues of MICRO-80 plus software cassettes plus Adventure, Space and strategy games, Light Pen 2 \$900.00
Sound off (value \$100's) freight free

G. Jackson,
P.O. Box 1012,
Nhulunbuy, N.T.
Ph. (089) B7-29B3

MICROTEK MT32 with 32K memory, an Exatron Stringy Floppy and 15 wafers. All \$500.00
excellent condition the lot

Brian J. Fillery,
Box 1790, G.P.O.
Brisbane, Q. 4001.

SYSTEM-80 L2/16K with over \$100 in programs, incl. BMON, RPN Calc. Sargon 3.5 Chess, Space Invaders, 70 Maths/Finance Programs. Excellent condition \$680.00
SERIAL PRINTER INTERFACE \$40.00

G. Hobbes,
42 Nareen Pde. NARABEEN, NSW. 2101.
Ph. (02) 913-2264

Tandy QUICK PRINTER II. 120 LPM Upper/Lower case on 2-3/B" aluminised paper. As new and 4 rolls of paper incl. \$200.00

INTERFACE to suit System-80 \$40.00

TEXAS INSTRUMENTS T159 Calculator & PC100C Thermal printer. The calculator has a card reader, 960 program steps & up to 100 memories. I have 2 extra plug in modules (each with about 22 programs) & some other software for it \$300.00

Dennis Bareis,
2B6 Lennox Street,
Maryborough, Q. 4650.
Ph. (071) 22-1699

***** BETTER BASIC PROGRAMMING - PART 4 *****
by Rod Stevenson

***** SCREEN FORMATTING *****

This particular episode of the series will be relatively short - not because it is unimportant, but because it is really only here to emphasise some of the points found in the Level II manual. In fact, there is not much to be said that can't be found in the manual and, after all, screen format is really simply a matter of trial and error - the results of a particular print format statement can be tried out until the desired result is obtained, then incorporated into the program.

While on the subject of trial and error, I should point out that this technique applies to many doubtful situations - that is, doubtful in the sense that when trying in a program, you can't think of the correct way. Yes, I know you ought to draw flow charts and write pseudocode and program on paper first, but how many do? I must admit that if a flow chart is required, I draw it after the program runs and is fully debugged! If I can't recall the exact syntax required for a simple statement, I often try it as a stand-alone line and run the line on its own to find out. An example that comes to mind is whether AND/OR is required, and if brackets will help, where to put them. Perhaps I should be ashamed to admit this (technique?) here, but I consider honesty to be more important than building a false image for ourselves!

The reason I have limited this article to screen formatting is the variation in the capabilities of printers and my assumption that possessors of printers will be willing to experiment on their own. As you will see, much of what I am to say on formatting is little more than sensible; the same applies to printer formatting. While allegedly some printers won't TAB past 64, there are various routines about to solve this lack - an example is: LPOINT STRING\$(T-PEEK(16539),32), where T is the required TAB and TAB must have been used previously in the program to provide a value to be held by the memory location 16539.

OF ? - I am constantly amazed at the otherwise knowledgeable computerists who don't know (why else wouldn't they use) the question mark as an abbreviation for PRINT. Not for LPRINT though.

COMMA - Used to advance to the next print zone before printing the current data. More than one can be used together to advance by more than one print zone.

SEMI COLON - Most commonly used to suppress the line feed after a print statement. The effect is that strings will print hard up after the previous one but numeric variables will have a space inserted before them (this is to allow for a minus sign should one be required).

I feel the semi colon is the most used print modifier because it is so useful. There is a need for care though - if the print statement will not always be followed by the same thing (because of branches in the program logic), be sure to check out that all possibilities will print in an acceptable manner.

TAB - A simple way of spacing your printout across the screen - up to 63 tab positions in one line mean there is no need to print a lot of blanks to get over to where you want your message to appear. Although there are theoretically up to 255 tab positions, the use of them becomes a little academic because it is difficult to visualise what is going to happen after you go past 64 and, if you want to go down to the next line, it would seem of more practical use to go to the next line before you start tabbing across the screen.

PRINT@ - A quite useful way of getting your message to be exactly where you want it to be on the screen. A map of the screen showing all PRINT@ positions (0 - 1023) is in the Level II manual. Or just use the trial and error method!

It's a good idea to follow your PRINT@ message with a semi colon, which will stop the cursor at the end of the message without wiping out whatever was on the screen previously. This means you can use PRINT@ to modify a screen full of information without disturbing anything other than the part changed. While INPUT@ won't work, you can PRINT@ a null string (" "), followed by a semi colon, colon, then INPUT just where it's required on the screen, and I do suggest a little thought on this technique, to make the screen display a little more orderly. Often it's useful to have all input at the same position on the screen, with previous INPUT being erased by the current input message, or by PRINT@X,CHR\$(30), which will erase the previous input line, then PRINT@X," ";:INPUTIN\$. This will mean input is always in the same screen position, and if the error checking routine (as detailed last episode) rejects the input, some error message can be printed there too, instead of disrupting screen format. Some of the other cursor control codes (found in the Level II manual) will be useful in dealing with input. Control - 29 will move the cursor to the beginning of the current line, 28 will move the cursor to the start of the screen but without clearing it, 24 - 27 will move the cursor in any of four directions; of course, any of these can be printed as multiples by STRING\$.

PRINT USING - Probably the most powerful, but also the most formidably complex to remember. I suggest referring to the Level II manual for full details and then experimenting until familiarity comes. But I do want to detail two of the most common uses which I use repeatedly. \$###.### will give dollars and cents format, with the cents rounded up or down. ### will give an output of three spaces (even if only one figure is printed). Both these are extremely useful in tabular output, and as they are right justified, make for tidiness. The PRINT USING format can be specified

at the beginning of a program, such as `PR$="####"`, so that whenever that particular format is required, `PRINT USING PR$` will produce it. Obviously a number of such statements can allocate a variety of `PRINT USING` formats, with the particular one being specified where required.

`PRINT USING` will override the space normally left in front of a number for a sign but allowance can be made for a sign - look in the manual for full details. So much is possible with `PRINT USING` that it is almost too much to cope with. Readers will no doubt recall that in the Array article, I used `PRINT USING` to format the output of the cross lotto selection numbers, so they could be in neat columns arranged just as required for aesthetic beauty.

CONCLUSION

As I said at the start, this episode is relatively brief but I do urge experimentation. I see little point in providing all the possibilities here when the reader will benefit more by actual use. For example, `PRINTTAB(30)LEST$(A$,4)` will work, but what about experimenting to determine how to combine (or simulate combinations) the separate statements of `PRINT@`, `PRINT USING`, `TAB`, `COMMA`, `SEMI COLON`?

Next instalment will be the promised (threatened?) one on assembly/machine language. Or to be more accurate, the combining of it with BASIC. I hope that, by now, those interested in this subject will have procured a copy of Bardon's book "TRS-80 Assembly Language Programming", available from Tandy, and will at least have opened it! because I don't intend starting at "grass roots" level. A book I also recommend is the Scelbi publication, Z80 Instruction Handbook as a reference handbook for the opcodes rather than puzzling one's way through the editor/assembler manual with its reprint of Zilog's handbook. This book is written in plain English with an understandable explanation of the instructions. It is a reference book I find goes well with Bardon's book explaining assembly language. The book has an index of codes in the back so that when you are wondering whether there is an instruction for the particular operation you want to do and cannot remember the more than 700 codes that the Z80 has, you can have a look in the index, find out if there is one, and then refer to the page given to find more detailed information about it.

- 0000000000 -

***** REVERSING THE ROLE OF PRINT AND LPRINT STATEMENTS ON THE TRS-80 *****

Reader, T.K. Nilsson, sent us an extract from the U.S. magazine *ELECTRONICS* which explains how to overcome the problem of your TRS-80 hanging-up if you run a program which contains `LPRINT` statements.

To direct `LPRINT` from the printer to the screen, type in:-

```
POKE16422,88:POKE16423,04 ENTER
```

In this mode, `PRINT` will work as expected and all data which would normally be sent to the line-printer will be directed to the screen.

To return to normal mode

```
POKE16422,141:POKE16423,05
```

If you would like to direct all `PRINT` statements to the line-printer instead of the screen so that you can obtain a hard copy of the program display, type in:-

```
POKE16414,141:POKE16415,04 ENTER
```

The printer will now respond to all outputs that would normally appear on the screen. It also echoes the keyboard.

To return to normal mode, type in:-

```
POKE16414,88:POKE16415,04.
```

One other useful piece of information concerns the line-printer status. If the line printer is out of paper then `PEEK14312` will equal 127. If it is not out of paper then `PEEK 14312` will equal 63. Incorporating this check into a BASIC program just before using an `LPRINT` statement will guard against system hang-ups.

- 0000000000 -

***** INPUT/OUTPUT *****

From: George Skorbek - Glen Waverley, Victoria.

I have encountered a problem associated with the CLEAR command and string manipulation. I have enclosed a sample program which will exhibit unexpected delays. Even the command PRINT FRE("X") takes a lot of time and I am curious as to why. If you can give me an explanation, I would be very grateful.

To conserve space, we have not reproduce the program sent in. It uses a CLEAR statement and involves the manipulation of a considerable number of String variables.

Your program has fallen victim to the "garbage collection" routine in Microsoft's BASIC interpreter. Garbage collection occurs when the string space you CLEARED, fills up, at least when the interpreter first looks at it. During the execution of a program, the data associated with most string variables is stored in the String Space allocated by the CLEAR statement. Every time you change the value of a string variable, another set of data is stored in the string space and the first set (i.e. its old value) is abandoned but still occupies space. For example: if your program first assigns the value "HELLO" to A\$ then later assigns it the value "PLEASE ENTER YOUR NAME", both "HELLO" and "PLEASE ENTER YOUR NAME" will be stored in the string space. Once the value of A\$ has been changed, HELLO is abandoned and irrecoverable as far as the program is concerned, but it still occupies the string space allocated to it.

String space is allocated to active strings, starting from the top of available memory (Memory size address) and working down. The BASIC interpreter maintains pointers to the "next available string location" and to "top of stack" (which is immediately below the last point in memory CLEARED. When a string is to be stored, the number of bytes required is subtracted from "next available" and the result is compared to "top of stack". When the comparison shows that space is available, the string is stored, the "next available" pointer is updated and program execution continues. However, when this test fails, the "garbage collector" routine is called upon.

The garbage collector first analyses all the string space and identifies abandoned string data. It then compacts all active strings up to the top of available memory, updating the pointers in the variables table as it goes, finally adjusting the "next available pointer". During this time, the CPU appears to be locked up, it will not respond to any key inputs and you may think your program has crashed irretrievably. Be patient, eventually it will continue. In programs having a large number of active strings, garbage collection may take 20 - 30 seconds or even longer! The garbage collection time is roughly proportional to the square of the number of active strings, i.e. garbage collection will take 4 times as long in a program with 10 active strings as in one with 5 active strings. For example, a program in a TRS-80 Model III having 500 active strings will "lock-up" for about 38 seconds every time the garbage collector is invoked.

The best way to reduce this time is to minimise the use of the garbage collector itself. CLEAR as much space as you possibly can. That way string space is much less likely to fill up. You should also reduce the number of strings you use, this will speed up the garbage collection time, when it does occur. Another way to reduce the frequency of garbage collection is to avoid abandoning string data by using LSET, RSET or MID\$. E.g. instead of A\$=B\$ use MID\$(A\$,1,12)=B\$. The value of A\$ will not then be abandoned.

Z\$=A\$+B\$+C\$+D\$ is disastrous since the interpreter treats this as:

```
T1$ = A$+B$
T2$ = T1$+C$
Z$ = T2$+D$
```

Instead, use:

```
Z$ = STRING$(21,0)
MID$(Z$,1,5)=A$
MID$(Z$,6,5)=B$
MID$(Z$,11,5)=C$
MID$(Z$,16,5)=D$
```

I hope this explains the phenomenon, George, and that the above will help you reduce its impact - Ed.)

***** JOYSTICKS AND INPUT/OUTPUT PORTS FOR YOUR '80 - PART 2 *****
by Alan Dent.

Last month I briefly described an interface that I have designed. It allows you to directly connect up to eight analogue signals into an analogue to digital converter, plus up to six parallel 8 bit input/output ports. This month I will expand on my descriptions of the two types of LSI chips used in the design, as well as giving a complete circuit description of the interface. The circuit and the printed circuit board is designed around the TRS-80 MODEL 1 and is easily connected to the expansion bus via an edge connector and ribbon cable. The connection to the P/C board is via a 40 pin IC socket header on the other end of the ribbon cable. As the external expansion connector in a SYSTEM-80 and also the TRS-80 MODEL 3 are of a different configuration, the connections to the P/C board have to be soldered to the holes for the 40 way IC socket instead of using the convenient plug in header. It is hoped in the near future to overcome this slight inconvenience to non-MODEL 1 owners.

ANALOGUE TO DIGITAL CONVERTER

The analogue to digital converter used is a NATIONAL SEMICONDUCTOR chip, type ADC0809. It is a CMOS device running from a single +5 volt supply, designed specifically for microprocessor applications. The internal parts of the A/D chip can be grouped into several sections: these are shown in the block diagram of the chip. Firstly, the address decoder. The address code for the desired input is presented to the 3 bit address input, followed by a positive-going pulse on the Address Latch Enable pin. This latches the address into the decoder and the decoded output selects the required channel to be switched on, in the analogue multiplexer. The selected analogue input signal is then connected through to one input of the voltage comparator. Now a positive-going start pulse must be fed into the Start pin. The Successive Approximation Register (S.A.R.), is reset on the positive edge of this pulse and the conversion sequence is initiated on the negative edge. The S.A.R. now, via the switch tree, sends successive voltage taps from the reference voltage resistor ladder, to the other input of the voltage comparator. It does this by sending an 8 bit binary word which is decoded into 1 of 256 switch combinations to select the voltage tap along the 256 resistors in the ladder. The S.A.R. continues this until a voltage match is achieved. At this point the Control and Timing section latches the 8 bit word that the S.A.R. used to achieve the voltage match, into the Tri-state Output Latch and then signals the End Of Conversion by raising this pin to a logic HI. The Tri-state output is then enabled under external control, and the digital value of the analogue signal, is placed on the DATA BUS.

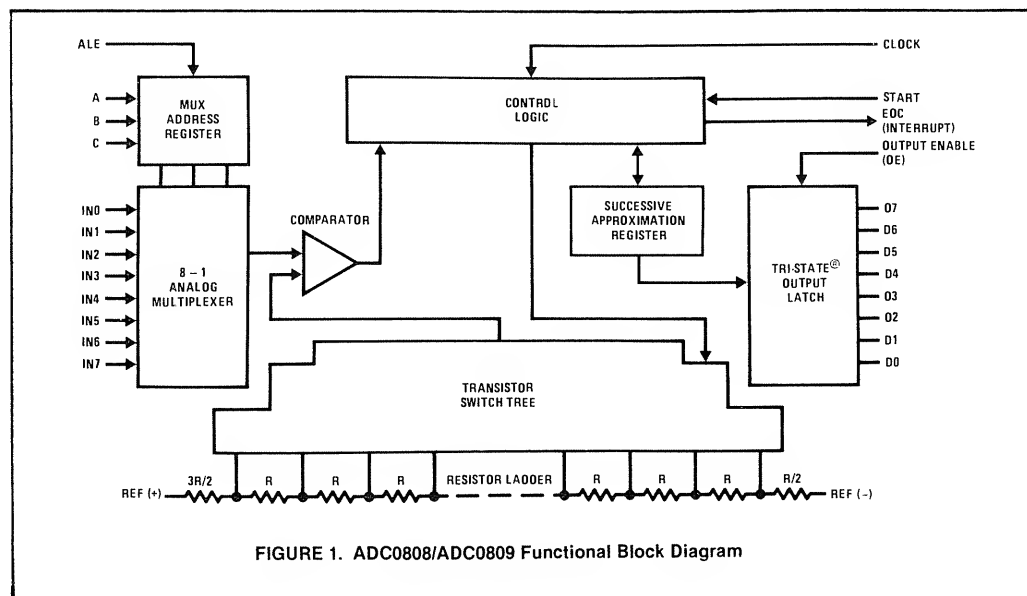
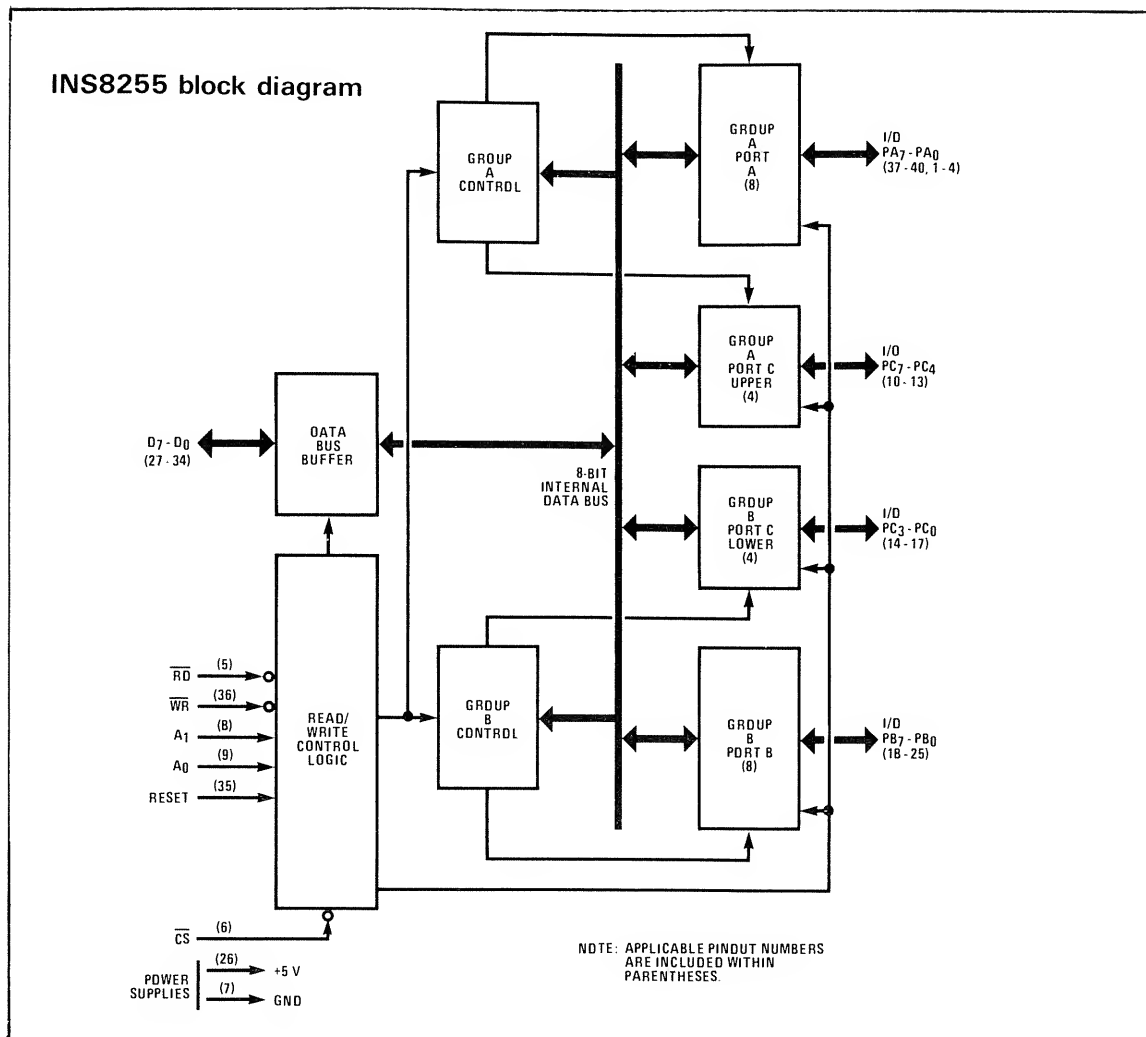


FIGURE 1. ADC0808/ADC0809 Functional Block Diagram

PARALLEL I/O PORT CHIP

The parallel I/O ports use an INTEL chip type 8255 called a PROGRAMMABLE PERIPHERAL INTERFACE. As you can see in the block diagram, it also has several internal sections. Starting at the Read/Write Control Logic, an incoming instruction is received and the appropriate action taken. As the PPI has an automatic power-on reset, the very first instruction must always be a write instruction into the group control registers. This control word then sets up the registers within the PPI and the port registers are programmed into that state until a new control word is sent to the control register or the system is reset. The Data Bus Buffer is a bi-directional buffer, controlled by the read/write logic. If a write instruction is received, the control logic first determines which register is being addressed, turns on the buffer to the receive mode and latches the word on the CPU data bus into the selected register. On a read instruction, the addressed port input data is latched and then presented to the internal bus, with the exception of port C which does not latch but only buffers the input data. The Data Bus Buffer then goes into the transmit mode and the data at the selected input port is placed on the CPU data bus. As can be seen in the block diagram, port C can also be split into two 4 bit ports and these can be programmed as either input or output independently of each other. Another very nice facility that port C has is its BIT SET and RESET capability. This allows the port output pin to be individually set or reset without changing the others; also, prior knowledge of the other pins is not required to change only one output pin. To change only one output bit on ports A & B, one would have to know the exact word that was in the output latch and write a new 8 bit word that only changed the required bit. This ability to set or reset one bit is an excellent one when using the port as a controller, and certainly makes the software much simpler. The direction of the I/O ports can be changed at any time simply by writing a new control word into the control register. Whenever this occurs, all output latches are reset, as well as the internal status flip-flops. That about covers the MODE 0 description; MODES 1 & 2 will not be covered here and must be obtained from the manufacturer's data books. Anybody wanting to use the other modes will probably be well into hardware already and more than likely has these books on hand.

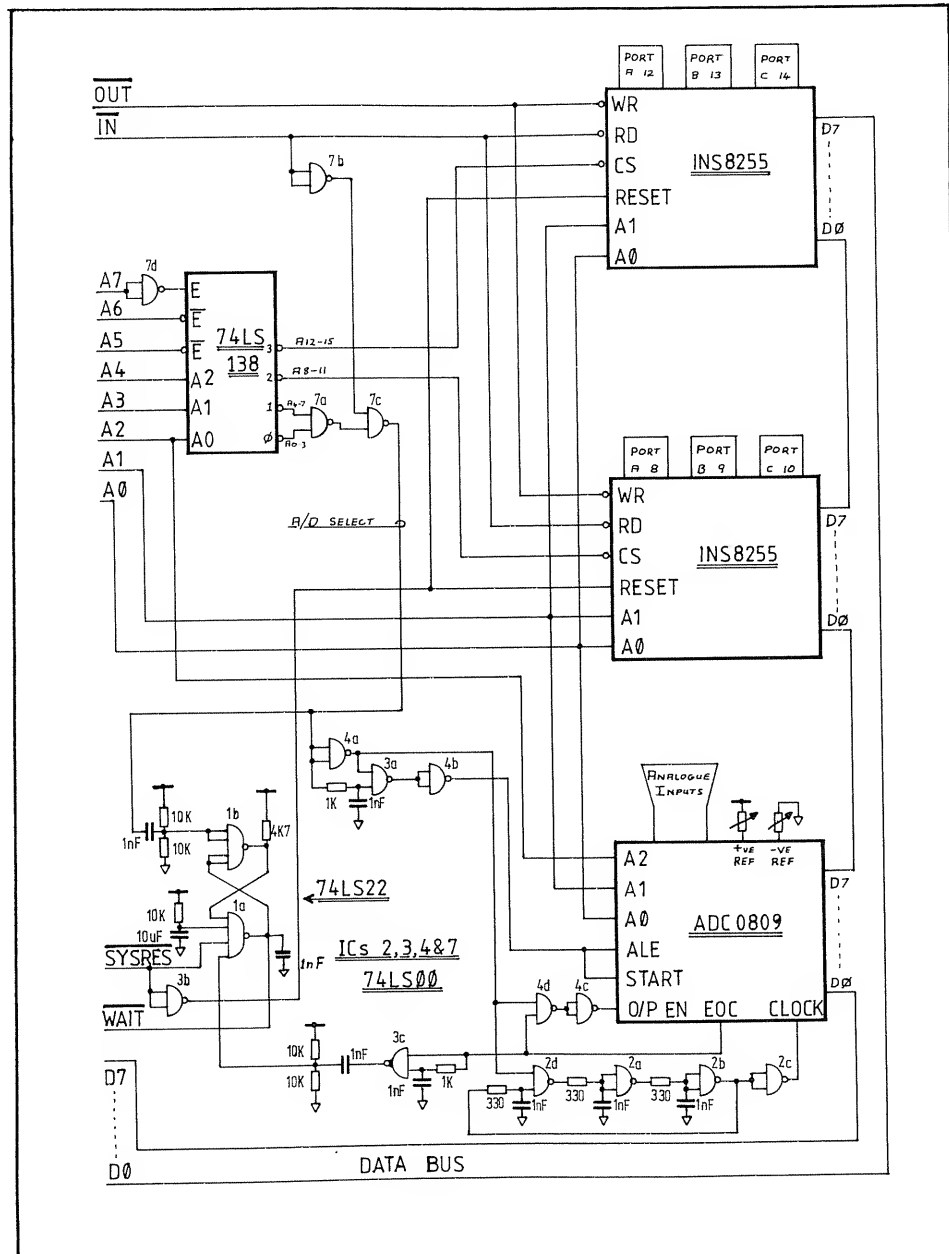


DETAILED OVERALL CIRCUIT DESCRIPTION

Now that you know how the LSI chips work internally, I hope that following the description of my circuit will be a little easier. The first section to be described is the address decoder. Address lines A2, A3 & A4 go to the address inputs of IC6, a 74LS138, one-of-eight decoder. The higher address lines A5, A6 and an inverted A7 are connected to the enable inputs of IC6 to ensure that it is only activated when the address is less than 32. As the lowest address line being used is A2, the individual decoded outputs, increment in steps of four. That means that output 0 represents addresses 0 - 3, and output 1 represents 4 - 7 etc. The 8 A/D converter inputs are decoded to appear at port addresses 0 - 7 and the PPI chips sit at port addresses 8 - 11 and 12 - 15. This means that the first two outputs of IC6 (0 & 1) must be combined to select the 8 inputs of the A/D; this is done in gate 7A, while the next two outputs (2 & 3) will individually select the two PPI chips. The remaining outputs are unused.

Assume a request is issued by the program for an INPUT from Port 0. First, the address on the address lines 0 - 7 goes to 00000000. Address lines A0, A1 and A2 are fed to the A/D converter Address Latch input. The Address Decoder chip IC6 is enabled by A5, A6 and an inverted A7 and the inputs to the decoder A2, A3 and A4 select output "0". The selected output goes L0 and forces the output of NAND gate 7A to go HI enabling one input of 7C. About 500 nanoseconds later the CPU issues a NOT IN command which is inverted by 7B and fed to the enabled gate 7C which then goes L0. The L0 output of 7C is the A/D SELECT signal. As the A/D conversion process takes about 100 microseconds, the CPU must be forced into the WAIT state while the conversion is taking place. This is done by pulling the input to 1B L0 which together with 1A form a SET/RESET flip flop. This L0 SETs the flip flop and the resulting L0 output of 1A pulls the CPU WAIT line L0 initiating the WAIT state. Gate 3A receives the A/D SELECT signal via 4A and also via a delay resistor capacitor network. This combination generates a negative going 1μs pulse on every negative going input. The pulse is inverted by 4B and is fed to the A/D chip. The positive going or leading edge of this pulse triggers the Address Latch Enable (ALE) input which latches the address "0" into the A/D address decoder. The decoded address then selects the analogue input "0" into the A/D converter. The same pulse also resets the S.A.R. inside the A/D. The negative going or trailing edge triggers the START conversion process within the A/D chip. The output of 4A also enables the A/D clock IC2 which runs at approx. 640 kHz. After approx. 100 μs the End Of Conversion (EOC) pin goes HI indicating that A/D conversion is finished. This signal is gated in 4D with the output of 4A and the output of 4D is then inverted by 4C which drives the Output Enable (OP EN) pin of the A/D. With the output enable pin HI the data lines are activated and the digital value of the decoded analogue input is placed on the DATA BUS. The EOC signal is also sent via a delay into 3C and the output of 3C is fed into 1A.

This RESETS the flip flop and releases the CPU from the WAIT state. The CPU then acquires the data on the DATA BUS and removes the NOT IN signal which then via 7B, 7C, 4A, 4B and 4C returns the A/D data lines to the high impedance state. The CPU then carries on executing its current program. Although pausing the CPU during the A/D conversion slows down ASSEMBLY program execution time, it is much easier than having to write special software routines to handle the task. I think this small sacrifice in speed is more than made up for with the convenience of being able to use standard PORT software and if accessing from BASIC, it is actually quicker than the INTERPRETER can interpret the BASIC code that would be required if the WAIT state were not used.



Now for a description of the parallel I/O port operation. We will assume the low address chip, which occupies A8 - A11 is being used. The control register for this chip will sit at PORT address A11. To set up this register the CPU outputs the address 00001011 on the address lines. The decoder then goes L0 on output 2 and selects the low address PPI chip by pulling its CHIP SELECT pin L0, at the same time A0 and A1 select register 3. The DATA BUSS then presents the control word to the chip and when the CPU issues its OUT command the data is written into the control register which sets up the internal registers. Assume an OUT 11,137 instruction was sent setting up PORTS A and B as output PORTS and PORT C as an input PORT, a following OUT 8,255 will latch 255 on the output pins of PORT A which will remain there until a different value is written to PORT B. An input command X=INP(10) will read the data at the input to PORT C and place it on the DATA BUSS. It is then moved into the CPU and assigned to the variable X.

For SYSTEM 80 users the NOT IN and NOT OUT signals are generated on the board by combining the I/O REQUEST signal with the READ and WRITE signals, as shown.

The power supply requires an external transformer; I personally think that the battery eliminator type of plug-pack power supplies are probably the most convenient. There is an onboard bridge rectifier, 2000uF filter capacitor and +5v voltage regulator chip. The regulator chip does not need a heatsink to handle the current consumed by the onboard components, including as many joysticks that you may wish to connect. At the same time, it has a very high reserve current capability. If this available extra current is called upon to drive some external connections such as TTL gates or LED displays etc., some form of heatsink is recommended.

In next month's concluding part of this series, I will be presenting construction details for the P/C board. It will include a copy of the P/C board artwork as well as a component stuffing diagram. The P/C board is a double-sided board but, to reduce the cost, it does not have plated through holes. There are about 75 pin through positions, none of which use components as the linking wire so that no components are soldered on the top of the board. This also allows the use of good quality sockets rather than the metal strip type of socket. Control words required for the programming of the PPI will be supplied as well as information on the BIT programming facility on PORT C. The addressing information for the PPI registers and ports will also be supplied. I hope to be able to present the necessary information in the form of a chart which should make programming and using the interface easier. Some sample programs will also be included to give some idea on how to program your '80 to use the interface. See you next month for the final part of this series.

- 0000000000 -

***** SOFTWARE SECTION *****

***** GUZINTA LI/4K *****
(C) Br. P. Van Eeken

The objective behind this program is to teach children to master the art of division, especially long division. The word divide or division is never mentioned in the program. Instead GUZINTA is used.

For the child, the objective is to destroy the alien space craft which appears at the beginning of the program. He sets up for himself a retaliation phasor for which he must supply the correct coordinates to destroy the alien. The coordinates are worked out in the form of a division sum. The force of the phasor is affected by the number of mistakes made and the difficulty of the sum. If the phasor lacks sufficient strength it will not penetrate the alien's defences and hence, the alien will escape. The program caters for 4 levels of difficulty.

```

1 REM                      *****
2 REM                      *          BY          *
3 REM                      * BR. P. VAN EEKEN *
4 REM                      *****
5 V=999:A#=GUZINTA
6 F.G=1T.5:C.:C=R.(11)+1:I=R.(C)+1:P.A.C*64+C*2;C;A#;C*I;
7 P."^";I;"TIMES":GOS.999:N.G
9 U=0:Q=263:K=0:A(13)=10:C.:P.A.212;"THE ";A#;
20 P." SPACE COMBAT GAME";A.404;"CHOSE A LEVEL OF COMBAT..."
24 C=1:P=2:R=3:I=4:REST.:F.D=532T.724S.64:REA.B#:P.A.D;B#:N.D
25 D.CUSHY,PLAIN SAILING,RESISTANCE,INSURMOUNTABLE,9,20,99,999
31 P.A.916;"YOUR CHOICE ";:I.A:P.A.916:F.G=1T.4:IFA=GG.40
34 N.G:G.31
40 F.W=1T.A:REA.B:N.W
45 C=R.(B):T=0:V=1:F.X=AT.OS.-1:A(X)=R.(9):T=T+V*A(X):V=V*10
46 N.X:A(A+1)=0:V=99:IF(A=3)*(C<10)+(A=4)*(C<100)G.45
50 F.X=1T.9:C.:P.A#;" SECURITY ALERT !!!":GOS.999:N.X
55 P.A.192;"COMMANDER, ";A#;" SCANNERS REPORT AN ALIEN PRESENCE
56 P."AT";C;A#;T;"SPACULES. PHASERS APPROACHING..."
60 Y=27:F.X=107T.127:S.(X,Y):IFX>122S.(X-3,Y)
61 Y=Y+1:N.X:F.X=101T.121:Y=Y-1:S.(X,Y):IFX<106S.(X+3,Y)
62 N.X:F.X=101T.104:S.(X,47):S.(X+23,47):N.X
63 F.X=108T.120:S.(X,43):N.X:GOS.650:S.(114,28):V=10:F.Y=0T.2
68 F.X=27T.OS.-1:S.(114,X):GOS.999:R.(114,X):N.X:N.Y
75 GOS.640:P."PHASERS MISSED TARGET ! ALIEN FORCE FIELD UP !
80 V=999:GOS.999:P.A.O:F.X=1T.7:P.:N.X
90 P.A.O;"RETALIATION PHASER LAUNCHING...":P=0:GOS.980
96 GOS.997:P."PLEASE COMPUTE PHASER CO-ORDINATES COMMANDER";

```

```

100 M=16:L=23+A*6:P.A.384;C;:P.A.391;:F.X=OT.A:P.A(X);:N.X
110 H=0;W=0
112 J=0:X=1:F.V=1T.A:X=X*10:N.V
113 F.I=OT.H:J=J+A(I)*X:X=X/10:N.I:B=A(I)
117 IFJ/10<>I.(J/10)G.120
118 J=J/10:G.117
120 IFC>JH=H+1:G.112
121 V=10:F.X=1T.3:IFJ<VG.125
122 Q=Q+3:V=V*10:N.X
125 N=I.(J/C):GOS.805:GOS.800:A(10)=0
130 P.A.64;"HOW MANY TIMES DOES";C;A#;J;:I.S:S.(114,P)
135 IFS<>NGOS.699:P.A.128;S;"X";C;"=";S*C;:G.130
139 P.A.Q;N;:Q=Q+3:GOS.700:Z=S*C:GOS.500:O=O+V=1:N=S*C
145 V=V*10:IFN>=VG.145
147 GOS.380:IFJ<VG.151
148 F.X=E-1T.A+2S.-1:A(X+1)=A(X):N.X:A(X+1)=O:E=E+1:V=V*10
150 A(10)=A(10)+3
151 D=453+K+U:P.A.D;"- ";V=J:X=0
152 IFV>OX=X+3:V=I.(V/10):G.152
153 IFN=OGOS.385
156 F.V=A+2T.E-1:IF0=XG.160
157 O=O+3:P.A(V);
160 N.V:N=J-S*C
170 GOS.700:P.A.64;J;"-";S*C;"=";:I.Z:S.(114,P)
175 IFZ<>NGOS.699:G.170
180 GOS.500:GOS.380:H=D+127+O+(W-G)-W
185 P.A.H;:F.X=A+2T.E-1:P.A(X);:H=H+3:N.X:IFB=OP."REMAINDER";
186 GOS.750:K=K+192:Z=J-S*C:IFZ=OG.200
191 A(11)=O:Z=1
192 IFJ-S*C>=ZA(11)=A(11)+1:Z=Z*10:G.192
193 Z=1:A(12)=O
194 IFS*C>ZA(12)=A(12)+1:Z=Z*10:G.194
196 IFA(12)<A(11)G.200
197 U=U+A(10)+3*(A(12)-A(11))
200 J=(J-S*C)*10+B:V=H:IF(J<10)*(S*C<10)U=U+3
206 V=V-64:IFV>448G.206
210 GOS.700:IFB=OG.600
215 F.X=OT.1:P.A.V;" ";GOS.999:P.A.V;B;:GOS.999:N.X
220 P.A.H;B;:I=I+1:B=A(I):G.125
380 IF(N/100<>I.(N/100))+(N=O)RET.
385 A(E)=O:E=E+1:IFN<999RET.
390 N=N/100:G.385
500 E=A+2:W=3:G=0
505 X=1
510 W=W+3:IFX*10>ZG.525
520 X=X*10:G.510
525 V=X:IFB>OG.530
526 G=G+3:V=V/10:IFZ<1OG=0
527 IFV>1G.526
530 F=I.(Z/X):A(E)=F:E=E+1:Z=Z-F*X:IF(Z=0)*(X=1)RET.
535 IFZ=OG.505
540 IFZ<X/10GOS.385
550 G.505
600 GOS.750:P.A.153;"ON COMPUTED CO-ORDINATES..."
601 V=O:F.X=PT.28:S.(114,X):GOS.999:R.(114,X)
602 IF(X>26)*(A(13)<9)S.(114,X)
603 N.X:P.A.O;
604 IF(A(13)=10)+(A>2)*(A(13)>8)G.607
605 P."PHASER FORCE TOO LOW & ALIEN ESCAPES !"
606 S.(114,27):V=2000:GOS.999:G.1
607 P."ALIEN FORCE FIELD PENETRATED & SHIP IMMOBILIZED."
608 R=2:GOS.640:F.X=1T.750:R.(R.(28)+99,R.(22)+27):N.X:G.1
640 F.V=108T.120:IFR=3S.(V,27):G.645
642 R.(V,27):R.(114,28)
645 N.V:RET.
650 F.Y=29T.35:F.X=113T.115:S.(X,Y):N.X:N.Y:RET.
699 A(13)=A(13)-1
700 P.A.64:P.:P.A.220;"PHASER AT FORCE":A(13):S.(114,P):RET.
750 GOS.700:P.A.135;"PHASER MANOEUVRES";:GOS.980:RET.
800 F.Y=16T.20:S.(13,Y):N.Y:F.Y=MT.M+9:S.(L,Y):N.Y:M=M+9
805 F.X=14T.L:S.(X,M):N.X:RET.
980 V=O:F.X=1T.S:GOS.985:N.X:R.(114,P):P=P+1
983 F.X=1T.S:GOS.985:N.X:RET.
985 R.(114,P):GOS.999:S.(114,P):GOS.999:RET.
997 P.A.O:P.A.O;:S.(114,P):RET.
999 F.W=OT.V:N.W:RET.

```


***** BIRTHDAY LI/4K *****
 (C) Br. P. Van Eeken

Like most of this author's programs, this one is designed for use in the classroom and it will not analyse a birthday until the student successfully solves the number of days between two dates that the computer supplies. It then tells the student the day of his birth, his age, the amount of time that he has slept, eaten, etc. and the amount of chickens, cattle, fruit, grain etc. he has eaten. It also supplies a year for suggested retirement!

```

1 REM *****
2 REM * BY *
3 REM * BR. P. VAN EEKEN *
4 REM *
5 REM *****
20 REM ADAPTED FROM THE PROGRAM 'WEKDAY'
30 REM CREATIVE COMPUTING VOL. 1.
70 C.:I."ENTER TODAY'S DATE IN THIS FORM: DD,MM,YY: ";A,B,C
105 M=0:IFC<100C=C+1900
106 IFM>=0G.110
107 REST.:F.X=1T.24:REA.Y:N.X:IFM=-1G.109
108 F.X=1T.A.(M+1):REA.D,E,F,Y,A#:N.X
109 REA.D,E,F,Y,A#:M=M-1:F=F+1900:G.225
110 V=C-1900:I."WHAT IS YOUR NAME ";A#:C.
111 P.A#: " TO HAVE YOUR BIRTHDAY ANALYSED, PLEASE TELL ME
112 D=R.(28):E=R.(11):F=R.(28):G=R.(12):IF(G<E)+(F<=D)G.112
115 P."HOW MANY DAYS THERE ARE BETWEEN THE";D: "/" ;E: "/" ;C: "AND"
116 P.T.(24): "----- THE";F: "/" ;G: "/" ;C
117 H=E:J=D:GOS.972:REST.:P:I:H=G:J=F:GOS.972:M=0
118 H=I-P-1
119 REM INSERT 'HELP' TO PASS QUESTION
122 P.A.448:P.A.448:"YOUR ANSWER ";:I:F:P.A.896:IFF=I-P-1G.130
123 P."SORRY ";A#: " , TRY AGAIN.":G.122
130 P."WELL DONE ";A#: " , ";:H=-1:F=0:M=1
131 REM TYPE 'HELP' TO DO GROUP... DATA AT LINES 983-987
132 I."ARE YOU A MALE OR FEMALE HUMAN ";Y:IFY=-1M=-1:G.106
133 IF(Y=M)+(Y=F)F.I=1T.12:REA.A(I):N.I:C.:G.135
134 P."RE-ENTER YOUR ANSWER PLEASE.":G.132
135 I."ENTER YOUR DATE OF BIRTH: DD,MM,YY ";D,E,F
201 IFF<100F=F+1900
225 I=I.((F-1500)/100):IFF-1582<0G.835
226 IFE=0C.:E.
235 K=I*5+(I+3)/4:J=I.(K-(I.(K/4)*7)):G=I.(F/100):H=I.(F-G*100)
255 K=H/4+H+D+A(E)+J:L=I.(K-(I.(K/7)*7))+1:IFE>2G.320
270 IFH=0G.305
275 F.X=1T.1:A(X)=I.(F-I.(K/4)*4):IFA(X)<>0G.320
277 N.X:IFL=0L=6
295 B=B-1:G.320
305 K=I-1:F.X=1T.1:A(X)=I.(K-(I.(K/4)*4):IFA(X)=0G.285
310 N.X
320 IFL=0L=7
325 P."THE ";D: "/" ;E: "/" ;F:
330 IF(C*12+B)*31+A<(F*12+E)*31+DT.360
335 IF(C*12+B)*31+A=(F*12+E)*31+DT.350
340 P." WAS A ";:G.365
350 P." IS A ";:G.365
360 P." WILL BE A ";
365 ONLG.366,376,386,396,406,810,426
366 P."SUNDAY ";:G.435
376 P."MONDAY ";:G.435
386 P."TUESDAY ";:G.435
396 P."WEDNESDAY ";:G.435
406 P."THURSDAY ";:G.435
426 P."SATURDAY ";
435 IF(F*12+B)*31+A=(F*12+E)*31+DG.720
436 P.A#:P.
440 N=C-F:Q=B-E:P=A-D:IFP>0G.480
470 Q=Q-1:P=P+30
480 IFQ>=0G.495
485 N=N-1:Q=Q+12
495 IFN<0G.850

```

```

496 IFZ<>-1T.500
497 N=N+1:P=0:Q=0:Z=0
500 P.T.(35);"YEARS";T.(45);"MONTHS";T.(55);"DAYS"
520 P."YOUR AGE:";T.(36);N;T.(46);Q;T.(56);P
521 G=N
525 Q=N*365+Q*30+P+I.(Q/2):S=Q:R=N:T=Q:U=P:V=F+65:W=.35
540 P."YOU HAVE SLEPT";:GOS.860:W=.17:P."YOU HAVE EATEN";
580 GOS.860:W=.23:IFR<4P."YOU HAVE PLAYED";:G.625
605 IFR>9P."YOU HAVE WORKED/STUDIED";:G.625
610 P."YOU HAVE PLAYED/STUDIED";
625 GOS.860:IFT<12T.630
627 R=R+1:T=0
630 P."YOU HAVE RELAXED";T.(36);R;T.(46);T;T.(56);U
631 IFG<4G.710
635 P."BASED ON AUSTRALIAN NATIONAL
636 P."AVERAGES, IN YOUR LIFETIME";:Z=150:B#=CATTLE.
637 GOS.960:P.:P."YOU HAVE EATEN...";:Z=850:B#=CHICKENS.
655 GOS.960:P.:Z=250:B#=SHEEP.
665 GOS.960:P.:Z=10:B#=HECTARE
685 GOS.960:IFW<>1P."S";
686 P." OF GRAIN.":P."YOU MAY RETIRE IN";:IFY=OV=V-5
690 P.V;:Z=20:GOS.960:IFW<>1P."S";
700 P." OF FRUIT & VEGES
705 IFM<OREST.:I."PRESS ENTER ";B#:G.106
710 P."NEXT HUMAN PLEASE, ";:REST.:G.106
711 G.110
720 P.:P."**** HAPPY BIRTHDAY **** ";A#;"!":Z=-1:G.440
810 IFD<>13P."FRIDAY ";:G.435
825 P."FRIDAY THE THIRTEENTH! ";:G.435
835 P."SORRY, CAN'T DO":REST.:G.106
860 X=I.(W*Q):N=I.(X/365):X=X-(N*365):O=I.(X/30):P=X-(O*30)
885 R=R-N:T=T-O:U=U-P:IFU>=OT.915
905 U=U+30:T=T-1
915 IFT<1T=T+12:R=R-1
930 P.T.(36);N;T.(46);Q;T.(56);P:RET.
960 W=I.(Z/(70*365)*S):Z=36:IFW>9Z=35:IFW>99Z=34
970 P.T.(Z);W;B.:RET.
971 D.31,28,31,30,31,30,31,31,30,31,30,31,0,3,3,6,1,4,6,2,5,0,3,5
972 I=O:F.X=1T.12:REA.A(X):IF(X=2)*(I.(V/4)=V/4)A(X)=29
975 IFH=XI=I+J:G.980
977 IFX<HI=I+A(X)
980 N.X:RET.
983 D.25,10,68,1,ANDREW,299,7,70,1,JOHN,99,12,69,0,ANN
986 D.18,6,70,0,BETTY,23,9,69,0,JULIE,7,8,56,1,HARRY
987 D.19,12,69,0,MICHELE,31,12,69,1,PAUL,29,4,63,1,TIM
988 D.0,0,0,0,END

```

***** BINARY CONVERTER LII/4K *****
(C) T. Fraser

The motivation for writing this program was Appendix 2 of the "TRS-80 ASSEMBLY LANGUAGE PROGRAMMING" book. That Appendix is all in binary numbers and our author felt more comfortable working in decimal or hexadecimal, hence this converter. (But to write it in BASIC !! - Ed.)

To use the program, simply type in an eight digit binary number, e.g. 10111000 and press ENTER/NEWLINE and the program will convert it to decimal and Hex and display the results on the screen. The program is very efficiently written and uses 300 bytes of memory.

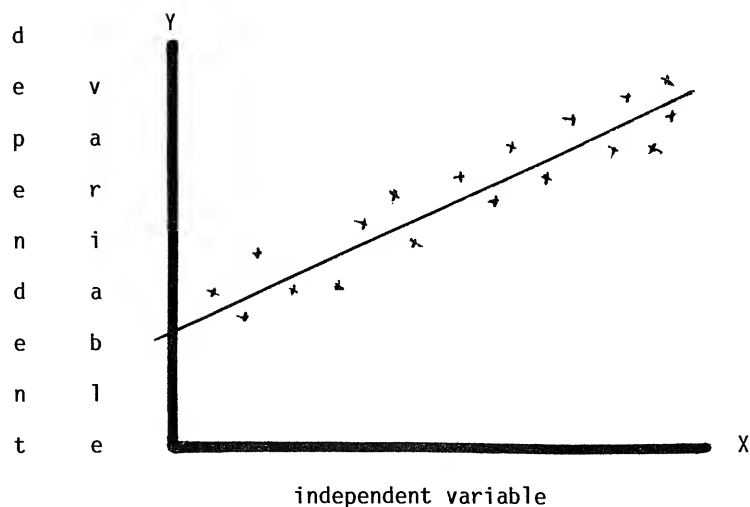
```

10 ' *****
    *   T. FRASER   *
    * GAYHURST RD  *
    *   KENWICK    *
    *   W.A.       *
20 ' *****
30 INPUT"INPUT BINARY NUMBER ";C#:W=LEN(C#):IFW<>8THEN30
40 BN=128:FORN=1TO8:D#=MID$(C#,N,1):A=ASC(D#)-48:IFA=1THENI=I+BN
   :BN=BN/2:A=0:NEXTELSEBN=BN/2:NEXT
50 J=INT(I/16):K=I-J*16:J=J+1:K=K+1
60 FORTZ=1TOJ:READA#:NEXT:RESTORE:FORTZ=1TOK:READB#:NEXT:RESTORE
70 DATA0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
80 CLS:PRINT"
  BINARY  NUMBER  ";C#
  DECIMAL  NUMBER  ";I;"
  HEX      NUMBER  ";A#:B#:RUN

```


***** LINEAR REGRESSION LII/4K *****
 (C) T.R. Jones

This program computes the linear relationship between two sets of data and follows the form:-



The mathematical model for the line is:-

$$Y = A + BX$$

Where A = the point on the Y axis where the line intercepts it.
 B = the slope of the line.

The program commences by asking for the number of data points required, (maximum 100), then prompts are given to enter firstly the X data and then the Y data for the number of data points selected. When all data has been entered it is displayed on the screen and the program asks if you wish to edit any of the data. If data is to be edited, the program will request the number of the data point to be edited and when given, will display the old values stored at that point, and then request the new values. After you have finished editing the data the program will display the MEAN for X and Y and the coefficient of linear correlation. The program will then ask if you require the residual table to be printed. If you do, this will then be displayed one screen full at a time. At the end of this list the program will enter a loop which, on each pass, will let you enter a value for X which will then display to the right, estimated values for Y.

Typical screen displays for the program are shown below:-

```

      LINEAR REGRESSION ANALYSIS
      LINEAR MODEL
      Y = A + BX
      TYPE (ENTER) TO CONTINUE ....?
  
```

```

      ENTER DATA
      NUMBER OF DATA POINTS - MAXIMUM IS 100
      *?
      DATA ENTERED
      X( 1 ) = 7      Y( 1 ) = 5
      X( 2 ) = 3      Y( 2 ) = 5
      X( 3 ) = 7      Y( 3 ) = 3
      X( 4 ) = 6      Y( 4 ) = 4
      X( 5 ) = 6      Y( 5 ) = 3
      X( 6 ) = 2      Y( 6 ) = 5
      X( 7 ) = 7      Y( 7 ) = 4
      X( 8 ) = 23     Y( 8 ) = 5
      X( 9 ) = 67     Y( 9 ) = 4
      X( 10 ) = 3     Y( 10 ) = 6
      TYPE (ENTER) TO CONTINUE ....?
  
```

X(11) = 76	Y(11) = 3
X(12) = 45	Y(12) = 7
X(13) = 4	Y(13) = 32
X(14) = 5	Y(14) = 76
X(15) = 4	Y(15) = 3
X(16) = 5	Y(16) = 6
X(17) = 4	Y(17) = 3
X(18) = 5	Y(18) = 6
X(19) = 4	Y(19) = 3
X(20) = 5	Y(20) = 6

TYPE <ENTER> TO CONTINUE?

```

EDIT DATA (Y/N)
DATA POINT TO BE EDITED? _
DATA POINT TO BE EDITED? 2
      X( 2 ) = 3
      Y( 2 ) = 5
NEW X( 2 ) =? 4
      NEW Y( 2 ) =?
      COMPUTING .....
THE EQUATION FOR THE REGRESSION LINE IS :
YHAT = 10.8959 - .107356 X
MEAN X = 14.4
MEAN Y = 9.35
COEFFICIENT OF LINEAR CORRELATION = -.138773

```

PRINT RESIDUAL TABLE (Y/N)				
POINT	X	Y	YHAT	RES
1	6.0000	4.0000	10.2518	6.2518
2	4.0000	4.0000	10.4665	6.4665
3	7.0000	3.0000	10.1444	7.1444
4	6.0000	4.0000	10.2518	6.2518
5	6.0000	3.0000	10.2518	7.2518
6	2.0000	5.0000	10.6812	5.6812
7	7.0000	4.0000	10.1444	6.1444
8	23.0000	5.0000	8.4267	3.4267
9	67.0000	4.0000	3.7031	-0.2969
10	3.0000	6.0000	10.5739	4.5739

TYPE CENTER TO CONTINUE ***

POINT	X	Y	YHAT	RES
11	76.0000	3.0000	2.7349	-0.2631
12	45.0000	7.0000	6.0649	-0.9351
13	4.0000	32.0000	10.4665	-21.5335
14	5.0000	76.0000	10.3591	-65.6409
15	4.0000	3.0000	10.4665	7.4665
16	5.0000	6.0000	10.3591	4.3591
17	4.0000	3.0000	10.4665	7.4665
18	5.0000	6.0000	10.3591	4.3591
19	4.0000	3.0000	10.4665	7.4665
20	5.0000	6.0000	10.3591	4.3591

SUM OF SQUARES OF DEVIATION FROM REGRESSION = 5324
ESTIMATE VALUES OF Y (Y/N)

X =? 9	Y = 9.92972
X =? 8	Y = 10.0371
X =? 7	Y = 10.1444
X =? 6	Y = 10.2518
X =? 5	Y = 10.3591
X =? 4	Y = 10.4665
X =? 3	Y = 10.5739
X =? 2	Y = 10.6812
X =? 1	Y = 10.7886
X =? 10	Y = 9.82237
X =? 11	Y = 9.71501
X =? 12	Y = 9.60766
X =? 13	Y = 9.5003
X =? 14	Y = 9.39294
X =? 15	Y = 9.28559
X =?	

DON'T BE HELD BACK BY AN ANTIQUATED DISK OPERATING SYSTEM MOVE UP TO

NEWDOS 80 \$149 incl. p&p

NEWDOS 80 is a completely new DOS for the TRS-80 SYSTEM 80. It is well-documented, bug free and increases the power of your system many times over. It is upward compatible with TRSDOS AND NEWDOS (ie TRSDOS and NEWDOS+ programs will run on NEWDOS 80 but the reverse is not necessarily so).

These are just a few of the many new features offered by NEWDOS 80.

- * New BASIC commands that support variable record lengths up to 4095 bytes long.
- * Mix or match disk drives. Supports any track count from 18 to 96. Use 35, 40, 77 or 80 track 5¼ inch mini disk drives, 8 inch disk drives OR ANY COMBINATION.
- * An optional security boot-up for BASIC or machine code application programs. User never sees "DOS-READY" or "READY" and is unable to "BREAK", clear screen or issue any direct BASIC statements, including "LIST".
- * New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- * Enhanced and improved RENUMBER that allows relocation of subroutines.
- * Create powerful chain command files which will control the operation of your system.
- * Device handling for routing to display and printer simultaneously.
- * MINIDOS -- striking the D, F and G keys simultaneously calls up a MINIDOS which allows you to perform many of the DOS commands without disturbing the resident program.
- * Includes Superzap 3.0 which enables you to display/print/modify any byte in memory or on disk.
- * Also includes the following utilities:
 - Disk Editor/Assembler
 - Disassembler (Z80 machine code)
 - LM offset — allows transfers of any system tape to Disk file — automatically relocated.
 - LEVEL 1 — Lets you convert your computer back to Level 1.
 - LVIDKSL — Saves and loads Level 1 programs to disk.
 - DIRCHECK — Tests disk directories for errors and lists them.
 - ASPOOL — An automatic spooler which routes a disk file to the printer whilst the computer continues to operate on other programs.
 - LCDVR — a lower case drives which display lower case on the screen if you have fitted a simple lower case modification.

DISK DRIVE USERS ELIMINATE CRC ERRORS AND TRACK LOCKED OUT MESSAGES FIT A PERCOM DATA SEPARATOR \$37.00 plus \$1.20 p&p.

When Tandy designed the TRS-80 expansion interface, they did not include a data separator in the disk-controller circuitry, despite the I.C. manufacturer's recommendations to do so. The result is that many disk drive owners suffer a lot of Disk I/O errors. The answer is a data separator. This unit fits inside your expansion interface. It is supplied with full instructions and is a must for the serious disk user.

MPI DISK DRIVES HIGHER PERFORMANCE — LOWER PRICE

MPI is the second largest manufacturer of disk drives in the world. MPI drives use the same form of head control as 8" drives and consequently, they have the fastest track-to-track access time available — 5msec! All MPI drives are capable of single or double-density operation. Double-density operation requires the installation of a PERCOM doubler board in the expansion interface.

As well as single head drives, MPI also makes dual-head drives. A dual-head drive is almost as versatile as two single-head drives but is much cheaper.

Our MPI drives are supplied bare or in a metal cabinet — set up to operate with your TRS-80 or SYSTEM 80. All drives are sold with a 90 day warranty and service is available through MICRO-80 PRODUCTS.

MPI B51 40 Track Single Head Drive.only \$349
MPI B52 40 Track Double Head Drive.only \$449

Prices are for bare drives and include p&p. Add \$10.00 per drive for a cabinet and \$60.00 for a power supply to suit two drives. 40 track drives are entirely compatible with 35 track drives. A 40 track DOS such as NEWDOS 80 is necessary to utilise the extra 5 tracks.

OVER 800 KILOBYTES ON ONE DISKETTE! WITH MPI 80 TRACK DRIVES

MPI 80 track drives are now available. The B91 80 track single-head drive stores 204 Kilobytes of formatted data on one side of a 5¼ inch diskette in single-density mode. In double-density mode it stores 408 Kilobytes and loads/saves data twice as quickly.

The B92 80 track dual-head drive stores 204 Kilobytes of formatted data on EACH side of a 5¼ inch diskette in single-density mode. That's 408 Kilobytes per diskette. In double-density mode, the B92 stores a mammoth 408 Kilobytes per side or 816 Kilobytes of formatted data per diskette. With two B92's and a PERCOM double, you could have over 1.6 Megabytes of on line storage for your TRS-80 for less than \$1500!!

MPI B91 80 Track Single Head Drive.only \$499
MPI B92 80 Track Dual Head Driveonly \$619

Prices are for bare drives and include p&p. Add \$10.00 per drive for a cabinet and \$60.00 for a power supply to suit two drives. Note: 80 track drives will not read diskettes written on a 35 or 40 track drive. If drives with different track counts are to be operated on the same system, NEWDOS 80 must be used.

CARE FOR YOUR DISK DRIVES? THEN USE 3M's DISK DRIVE HEAD CLEANING DISKETTES \$30.20 incl. p&p.

Disk drives are expensive and so are diskettes. As with any magnetic recording device, a disk drive works better and lasts longer if the head is cleaned regularly. In the past, the problem has been, how do you clean the head without pulling the mechanism apart and running the risk of damaging delicate parts. 3M's have come to our rescue with SCOTCH BRAND, non-abrasive, head cleaning diskettes which thoroughly clean the head in seconds. The cleaning action is less abrasive than an ordinary diskette and no residue is left behind. Each kit contains:

- 2 head cleaning diskettes
- 1 bottle of cleaning fluid
- 1 bottle dispenser cap

USE TANDY PERIPHERALS ON YOUR SYSTEM-80 VIA

SYSPAND-80 - \$97.50 incl. p&p

The SYSTEM-80 hardware is not compatible with the TRS-80 in two important areas. The printer port is addressed differently and the expansion bus is entirely different. This means that SYSTEM-80 owners are denied the wealth of economical, high performance peripherals which have been developed for the TRS-80. Until now, that is. MICRO-80 has developed the SYSPAND-80 adaptor to overcome this problem. A completely self-contained unit in a small cabinet which matches the colour scheme of your computer, it connects to the 50-way expansion part on the rear of your SYSTEM 80 and generates the FULL Tandy 40 way bus as well as providing a Centronics parallel printer port. SYSPAND-80 enables you to run an Exatron Stringy Floppy from your SYSTEM 80, or an LNW Research expansion interface or any other desirable peripherals designed to interface to the TRS-80 expansion port. Make your SYSTEM 80 hardware compatible with the TRS-80 via SYSPAND-80.

PROGRAMS BY MICROSOFT

EDITOR ASSEMBLER PLUS (L2/16K)

\$37.50 + \$1.20 p&p

A much improved editor-assembler and debug/monitor for L2/16K TRS-80 or SYSTEM 80. Assembles directly into memory, supports macros and conditional assembly, includes new commands-substitute, move, copy and extend.

LEVEL III BASIC **\$59.95 plus \$1.20 p&p**

Loads on top of Level II BASIC and gives advanced graphics, automatic renumbering, single stroke instructions (shift-key entries) keyboard debounce, suitable for L2/16K and up (Not Disk BASIC)

ADVENTURE ON DISK **\$35.95 plus \$1.20 p&p**

This is the original ADVENTURE game adapted for the TRS-80. The game fills an entire diskette. Endless variety and challenge as you seek to rise to the level of Grand Master. Until you gain skill, there are whole areas of the cave that you cannot enter. (Requires 32K One Disk)

BASIC COMPILER **\$208 plus \$2.00 p&p**

New improved version, the Basic Compiler converts Disk BASIC programs to machine code, automatically. A compiled program runs, on average, 3-10 times faster than the original BASIC program and is much more difficult to pirate.

UPGRADE TO 16K FOR ONLY \$30.00!!

MICRO-80's 16K MEMORY EXPANSION KIT HAS BEEN REDUCED IN PRICE EVEN MORE

Larger volume means we buy better and we pass the savings on to you. These are our proven, prime, branded 200 ns (yes, 200 nanosecond) chips. You will pay much more elsewhere for slow, 350 ns. chips. Ours are guaranteed for 12 months. A pair of DIP shunts is also required to upgrade the CPU memory in the TRS-80 — these cost an additional \$4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to instal them.

DISK DRIVE CABLES SUITABLE FOR ANY DISK DRIVES

DC-2 2 Drive Connector Cable \$39 incl. p&p
DC-4 4 Drive Connector Cable \$49 incl. p&p

DOUBLE THE SPEED AND CAPACITY OF YOUR DISK DRIVES

PERCOM DOUBLER ONLY \$220

plus \$2.00 p&p

Installing a Doubler is like buying another set of disk drives, only much cheaper!! The doubler works with most modern disk drives including:- MPI, Micropolis, Pertec, TEAC (as supplied by Tandy). The doubler installs in the TRS-80 expansion interface, the System-80 expansion interface and the LNW Research expansion interface in a few minutes without any soldering, cutting of tracks, etc. It comes complete with its own TRSDOS compatible double density operating system.

DOUBLE-ZAP II — DOUBLE DENSITY PATCH FOR NEWDOS 80

ONLY \$53.00 plus \$1.00 p&p

If you are using NEWDOS 80, then you also need DOUBLE-ZAP II on diskette. This program upgrades your NEWDOS 80 to double density with ADR (automatic density recognition). It retains all the familiar features, including the ability to mix and match track counts on the same cable. In addition, it gives NEWDOS 80 the ability to mix densities on the same cable, automatically. If you place a single density diskette in drive 0, say and a double density diskette in drive 1, Double-ZapII will recognise this and read/write to drive 0 in single density whilst at the same time it reads/writes to drive 1 in double density!

FLOPPY DOCTOR AND MEMORY DIAGNOSTIC (by MICRO CLINIC) \$29.95 plus 50c. p&p

Two machine language programs on a diskette together with manual which thoroughly test your disk drives and memory. There are 19 possible error messages in the disk drive test and their likely causes are explained in the manual. Each pass of the memory tests checks every address in RAM 520 times, including the space normally occupied by the diagnostic program itself. When an error occurs the address, expected data, and actual data are printed out together with a detailed error analysis showing the failing bit or bits, the corresponding IC's and their location. This is the most thorough test routine available for TRS-80 disk users.

BOOKS

LEVEL II ROM REFERENCE MANUAL

\$24.95 + \$1.20 p&p

Over 70 pages packed full of useful information and sample programs. Applies to both TRS-80 and SYSTEM 80.

TRS-80 DISK AND OTHER MYSTERIES

\$24.95 + \$1.20 p&p

The hottest selling TRS-80 book in the U.S.A. Disk file structures revealed, DOS's compared and explained, how to recover lost files, how to rebuild crashed directories — this is a must for the serious Disk user and is a perfect companion to any of the NEWDOS's.

LEARNING LEVEL II

\$16.95 + \$1.20 p&p

Written by Daniel Lien, the author of the TRS-80 Level I Handbook, this book teaches you, step-by-step, how to get the most from your Level II machine. Invaluable supplement to either the TRS-80 Level II Manual or the System-80 Manuals.

MORE AUSTRALIAN SOFTWARE

All programs designed to run on both the TRS-80 or the SYSTEM 80 without modification. Most programs include sound

TRIAD VOL 1 -- L2/16K

Cassette \$10.95 Disk \$15.95

+ 60c p&p

Three separate games which test your powers of memory and concentration. The programs combine graphic displays and sound:

SIMON-SEZ: Just like the electronic music puzzles on sale for more than \$20. Numbers are flashed on the screen and sounded in a sequence determined by the computer. Your task is to reproduce the sequence, correctly.

LINE?: Rather like a super, complicated version of noughts and crosses. You may play against another player or against the computer itself. But beware, the computer cheats!

SUPER CONCENTRATION: Just like the card game but with more options. You must find the hidden pairs. You may play against other people, play against the computer, play on your own, or even let the '80 play on its own.

TRIAD VOL 2 -- L2/16K

Cassette \$10.95 Disk \$15.95

+ 60c p&p

Remember those "NUMERO" puzzles in which you had a matrix of numbers (or letters) with one blank space and you had to shuffle the numbers around one at a time until you had made a particular pattern? Well, **SHUFFLEBOARD**, the first program in this triad, is just this, except that the computer counts the number of moves you take to match the pattern it has generated -- so it is not possible to cheat.

MIMIC is just like **SHUFFLEBOARD** except that you only see the computer's pattern for a brief span at the beginning of the game, then you must remember it!

In **MATCHEM**, you have to manoeuvre 20 pegs from the centre of the screen to their respective holes in the top or bottom rows. Your score is determined by the time taken to select a peg, the route taken from the centre of the screen to the hole and your ability to direct the peg into the hole without hitting any other peg or the boundary.

VISURAMA L2/16K

Cassette \$10.95 Disk \$15.95

+ 60c p&p

Two programs which give fascinating, ever-changing patterns on the screen.

LIFE is the fastest implementation of the Game of Life you will see on your '80. Machine language routines create up to 1200 new generations per minute for small patterns or up to 100 per minute for the full 128 x 48 screen matrix. Features full horizontal and vertical wraparound.

EPICYCLES will fascinate you for hours. The ever-changing ever-moving patterns give a 3D effect and were inspired by the ancient Greek theories of Ptolemy and his model of the Solar system.

EDUCATION AND FUN -- L1/4K, L2/16K

Cassette \$10.95 Disk \$15.95

+ 60c p&p

Written by a primary school teacher to make learning enjoyable for his pupils, there are five programs in both Level I and Level II to suit all systems:

BUG-A-LUG: a mathematics game, in which you must get the sum correct before you can move.

AUSTRALIAN GEOGRAPHY: learn about Australian States and towns, etc.

SUBTRACTION GAME: build a tower with correct answers.

HOW GOOD IS YOUR MATHS? Select the function (+, -, ÷ or X) and degree of difficulty.

HANGMAN: That well known word game now on your computer.

Recommended for children from 6 to 9 years.

COSMIC FIGHTER & SPACE JUNK -- L2/16K

Cassette \$10.95 Disk \$15.95

+ 60c p&p

Both programs have sound to complement their excellent graphics. In **COSMIC FIGHTER**, you must defend the earth against seven different types of alien aircraft. It is unlikely that you will be successful but you will have a lot of fun trying!

You mission in **SPACE JUNK** is to clean up all the debris left floating around in space by those other space games. It is not as simple as it sounds and space junk can be quite dangerous unless you are very careful.

SPACE DRIVE L2/4K & 16K

Cassette \$8.95 Disk \$13.95

+ 60c p&p

Try to manoeuvre your space ship through the meteor storms then land it carefully at the space port without running out of fuel or crashing. Complete with realistic graphics.

STARFIRE AND NOVA INVASION L2/16K

Cassette \$10.95 Disk \$15.95

+ 60c p&p

Both programs include sound to improve their realism.

STARFIRE seats you in the cockpit of an X-wing fighter as you engage in battle with the deadly Darth Vader's Tie-fighters. Beware of the evil one himself and may the Force be with you.

In **NOVA INVASION**, you must protect your home planet of Hiberna from the invading NOVADIANS. You have two fixed guns at each side of the screen and a moveable one at the bottom. Apart from shooting down as many invaders as possible, you must protect your precious hoard of Vitaminium or perish!

AIR ATTACK AND NAG RACE -- L2/16K

Cassette \$10.95 Disk \$15.95

+ 60c p&p

An unlikely combination of programs but they share the same author who has a keen sense of humour.

AIR ATTACK includes sound and realistic graphics. The aircraft even have rotating propellers! But they also drop bombs on you, so it's kill or be killed!

NAG RACE lets you pander to your gambling instinct without actually losing real money. Up to five punters can join in the fun. Each race results in a photo-finish whilst there is a visible race commentary at the bottom of the screen throughout the race. Happy punting!

FOUR LETTER MASTERMIND L2/16K

Cassette \$8.95 Disk \$13.95

+ 60c p&p

There are 550 four-letter words from which the computer can make its choice. You have 12 chances to enter the correct word. After each try, the computer informs you of the number of correct letters and those in the correct position. You can peek at the list of possible words but it will cost you points. Makes learning to spell fun.

MUSIC IV -- L2/16K

Cassette \$8.95 Disk \$13.95

+ 60c p&p

Music IV is a music compiler for your '80. It allows you to compose or reproduce music with your computer that will surprise you with its range and quality. You have control over duration (full beat to 1/16 beat) with modifications to extend the duration by half or one third for triplets. Both sharps and flats are catered for as are rests. Notes on whole sections may be repeated. The program comes with sample data for a well-known tune to illustrate how it is done.

SAVE 00's SAVE 00's***SAVE 00's*** MICRO-80 EXPANSION INTERFACE ***

MICRO-80's expansion interface utilises the proven LNW Research Expansion board. It is supplied fully built up and tested in an attractive cabinet with a self contained power supply, ready to plug in and go. The expansion interface carries MICRO-80's full, no hassle, 90-day warranty.

Features include:- • Sockets for up to 32K of memory expansion • Disk controller for up to 4 disk drives • Parallel printer port • Serial RS232C/20mA I/O port • Second cassette (optional)

The expansion interface connects directly to your TRS-80 L2/16K keyboard or, via SYSPAND-80 to your SYSTEM-80/VIDEO GENIE
Prices: HD-010-A Expansion Interfaces with 0 K : \$499.00 HD-010-B Expansion Interfaces with 32K : \$549.00 HD-011 Data separator fitted (recommended) : add \$29.00 HD-012 Dual cassette Interfaces fitted : add \$19.00

The MICRO-80 Expansion Interface is also available in kit form.

Prices: HD-013 Kit consisting of LNW Research PC board and manual, ALL components including cabinet & power supply : \$375.00
HD-011 Data separator for above \$25.00 HD-013 Dual cassette Interface kit : \$15.00

**TURN
THIS**

**into
this**

for \$49.00 plus \$2.00 p & p

A choice of upper and lower case display is easier to read, gives greater versatility.

The Micro-80 lower case modification gives you this facility, plus the symbols for the 4 playing-card suits for \$49.00 + \$2.00 p. & p.

The Micro-80 modification features true below-the-line descenders and a block cursor.

Each kit comes with comprehensive fitting instructions and two universal lower-case drive routines on cassette to enable you to display lower case in BASIC programs.

The driver routines are self-relocating, self-protecting and will co-reside with other machine language programs such as Keyboard-debounce, serial interface driver programs etc.

Both programs give your TRS-80™ Model I or System 80™ an optional typewriter capability, i.e. shift for upper case.

The second programme also includes Keyboard-debounce and a flashing cursor.

You fit it. Or we can.

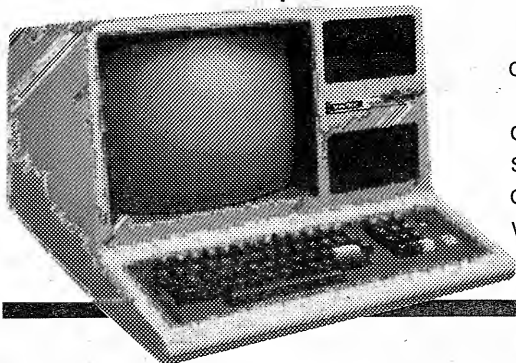
Fitting the modification requires soldering inside the computer. This should only be carried out by an experienced hobbyist or technician.

If you are at all dubious, a fitting service is available in all capital cities for only \$20.00

A list of installers is included with each kit.

Save \$120 now.

**ADD A DISK DRIVE TO YOUR TRS-80™ MODEL III
FOR ONLY \$875.00 OR ADD TWO FOR ONLY \$1199.**



The Micro-80 disk drive upgrade for the TRS-80™ Model III contains the following high quality components:

1 or 2 MPI 40-track single head disk drives, 1 VR Data double-density disk controller board and 1 dual drive power supply plus all the necessary mounting hardware, cables and comprehensive fitting instructions, which can be carried out with a minimum of fuss by any average computer owner.

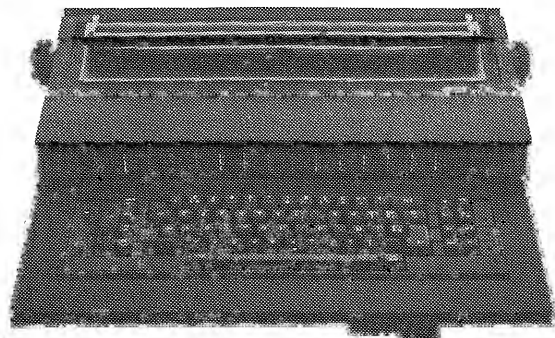
Fitting service is available for \$25.00 in most capital cities.

Daisy Wheel Typewriter/Printer

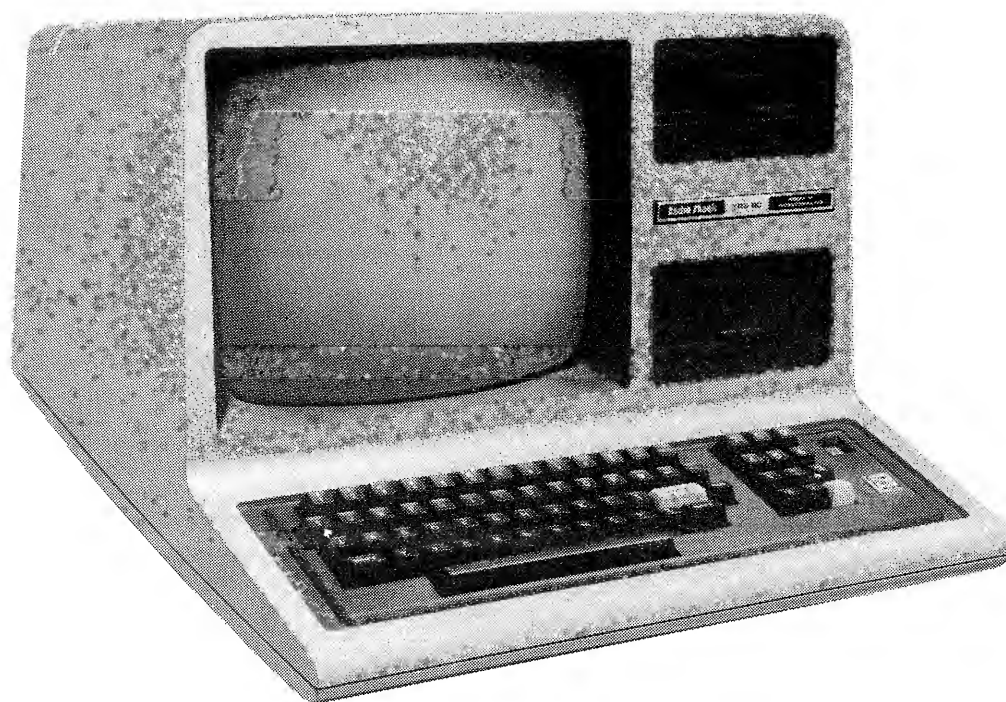
MICRO-80 has converted the new OLIVETTI ET-121 DAISY WHEEL typewriter to work with the TRS-80 and SYSTEM 80 or any other microcomputer with a Centronics parallel port (RS 232 serial interface available shortly). The ET-121 typewriter is renowned for its high quality, fast speed (17 c.p.s.), quietness and reliability. MICRO-80 is renowned for its knowledge of the TRS-80/SYSTEM 80 and its sensible pricing policy. Together, we have produced a dual-purpose machine:- an attractive, modern, correcting typewriter which doubles as a correspondence quality Daisy-wheel printer when used with your micro-computer.

How good is it? - This part of our advertisement was typeset using an ET-121 driven by a TRS-80. Write and ask for full details.

ONLY \$2049 INC. S.T.



1.4 MEGABYTES ON LINE + 48K RAM
for \$3800 incl. Sales Tax



MICRO-80's
MODEL 380 +

MICRO-80 has equipped the TRS-80 with two high reliability dual-head 80 track mini-floppy disk drives made by MPI, one of America's leading mini-disk drive manufacturers.

This turns the mild-mannered Model 3 into a powerhouse able to handle the most difficult business programs. The TRS-80 is one of the best-supported microcomputers in the world. MICRO-80 has been supporting the TRS-80 in Australia for 18 months and is one of Australia's leading dealers in MPI disk drives.

2.8 MEGABYTES FOR \$5300 incl. Sales Tax

If you need even more file space you can add MICRO-80's external dual-drive cabinet enclosing two more dual-head 80 track drives for an additional \$1500.

COMPUTER PRICES

MODEL 340

2 40 TRACK SINGLE HEAD DRIVES GIVING
350K FORMATTED STORAGE, 48K RAM

\$2990 INCL. SALES TAX

MODEL 340 +

2 40 TRACK DUAL-HEAD DRIVES GIVING
700K FORMATTED STORAGE, 48K RAM

\$3350 INCL. SALES TAX

MODEL 380

2 80 TRACK SINGLE HEAD DRIVES GIVING
700K FORMATTED STORAGE, 48K RAM

\$3350 INCL. SALES TAX

MODEL 380 +

2 80 TRACK DUAL-HEAD DRIVES GIVING
1.4 MEGABYTE FORMATTED STORAGE, 48K RAM

\$3800 INCL. SALES TAX

350K SYSTEM

MODEL 340, EPSON MX-80 PRINTER
NEWDOS 80 DISK OPERATING SYSTEM

\$4070 INCL. SALES TAX

700K SYSTEM (40 Track)

MODEL 340 + , EPSON MX-80 PRINTER
NEWDOS 80 DISK OPERATING SYSTEM

\$4429 INCL. SALES TAX

700K SYSTEM (80 Track)

MODEL 380, EPSON MX-80 PRINTER
NEWDOS 80 DISK OPERATING SYSTEM

\$4429 INCL. SALES TAX

1.4 MEGABYTE SYSTEM

MODEL 380 + , EPSON MX-80 PRINTER
NEWDOS 80 OPERATING SYSTEM

\$4880 INCL. SALES TAX

2.8 MEGABYTE SYSTEM

MODEL 380 + , DUAL EXTERNAL DRIVES,
MX-80 PRINTER, NEWDOS 80 OPERATING SYSTEM

\$6380 INCL. SALES TAX



EXATRON STRINGY FLOPPY — \$372.50 Incl. P&P

All Exatron Stringy Floppies sold by MICRO-80 include the special chained version of **HOUSEHOLD ACCOUNTS**, developed by Charlie Bartlett. When used on the ESF, this program is powerful enough to perform many of the accounting functions in a small business. Remember, the ESF comes complete with a comprehensive manual, a 2 way bus-extender cable, its own power supply and 10 wafers of mixed length. One wafer contains the Data Input/Output program and another the **HOUSEHOLD ACCOUNTS** program.

CAN'T MAKE UP YOUR MIND ABOUT THE ESF?

Then send in \$5.00 for a copy of the manual. We will refund your \$5.00 IN FULL when you purchase an ESF.



All prices subject to change without notice. Prices are F O B Adelaide. All computers and systems carry MICRO-80's 90-day Warranty covering parts and labour.

SOFTWARE BY AUSTRALIAN AUTHORS

All our software is suitable for either the SYSTEM 80 or the TRS-80

NEW SOFTWARE FROM MICRO-80 PRODUCTS BUSINESS PROGRAMS

MICROMANAGEMENT STOCK RECORDING SYSTEM (L2/16K)

Cassette version. \$29.95 + \$1.00 p&p
Stringy Floppy version. \$33.95 + \$1.00 p&p

This system has been in use for 9 months in a number of small retail businesses in Adelaide. It is therefore thoroughly debugged and has been tailor made to suit the requirements of a small business. MICROMANAGEMENT SRC enables you to monitor the current stock level and reorder levels of 500 different stock items per tape or wafer. It includes the following features:—

- Add new items to inventory
- Delete discontinued items from inventory
- List complete file
- Search for any stock number
- Save data to cassette or wafer
- Load data from cassette or wafer
- Adjusts stock levels from sales results and receipt of goods
- List all items requiring reordering

We can thoroughly recommend this program for the small business with a L2/16K computer.

SCOTCH BRAND COMPUTING CASSETTES

Super-quality personal computing cassettes.

C-10 pack of 10 \$26.00 incl. p&p
C-30 pack of 10 \$28.00 incl. p&p

UTILITIES

S-KEY by Edwin Paay **\$15.95 plus 50c. p&p**

S-KEY is a complete keyboard driver routine for the TRS-80 and becomes part of the Level II basic interpreter. With S-KEY loaded the user will have many new features not available with the standard machine.

S-KEY features:

- * S-KEY provides an auto-repeat for all the keys on the keyboard. If any key is held down longer than about half a second, the key will repeat until it is released.
- * Graphic symbols can be typed direct from the keyboard, this includes all 64 graphic symbols available from the TRS-80/SYSTEM 80.
- * S-KEY allows text, BASIC commands and/or graphics to be defined to shifted keys. This makes programming much easier as whole commands and statements can be recalled by typing shift and a letter key.
- * Because S-KEY allows graphics to be typed directly from the keyboard, animation and fast graphics are easily implemented by typing the appropriate graphics symbols directly into PRINT statements.
- * S-KEY allows the user to LIST a program with PRINT statements containing graphics, properly. S-KEY does this by intercepting the LIST routine when necessary.
- * S-KEY allows the user to list an updated list of the shift key entries to the video display or line printer.
- * S-KEY can be disabled and enabled when required. This allows other routines which take control of the keyboard to run with S-KEY as well.

Each cassette has TRS-80, DISK and SYSTEM 80 versions and comes with comprehensive documentation.

BMON by Edwin Paay **\$19.95 plus 50c. p&p**
THE ULTIMATE HIGH MEMORY BASIC MONITOR L2/16-48K

Our own personnel refuse to write BASIC without first loading this amazing machine language utility program into high memory! BMON Renumbers; Displays BASIC programs on the screen while they are still loading; tells you the memory locations of the program just loaded; lets you stop a load part-way through; merges two programs, with automatic renumbering of the second so as to prevent any clashes of line numbers; recovers your program even though you did type NEW: makes one program invisible while you work on a second (saves hours of cassette time!); lists all the variables used in the program; makes SYSTEM tapes; lets you Edit memory directly . . . the list goes on and on. Cassette comes with 16K, 32K and 48K versions, ready to load. Can anyone afford NOT to have BMON?

EDUCATIONAL

RPN CALCULATOR (L2/16K & 32K)
\$14.95 \$50c. p&p

Give your computer the power of a \$650 reverse polish notation calculator with 45 functions and selectable accuracy of 8 or 16 digits. The main stack and registers are continuously displayed whilst the menu is always instantly accessible without disturbing any calculations or register values. The cassette comes with both the 16K and 32K versions, the latter giving you the additional power of a programmable calculator. Comes with a very comprehensive 15 page manual, which includes instructions to load and modify the 32K programmable version to run in 16K. Whether for business or pleasure, this package will prove invaluable, and turn you '80 into a very powerful instrument.

GAMES

MICROPOLY (L2/16K) **\$8.95 + 60c p&p**

Now you can play Monopoly on your micro. The old favourite board game has moved into the electronic era. This computer version displays the board on the screen, obeys all the rules and, best of all, the banker does not make mistakes with your change!

CONCENTRATION (L2/16K) **\$8.95 + 60c p&p**

Another application of supergraphics. There are 28 "cards" displayed on the screen, face down. Players take it in turn to turn them over with the object of finding matching pairs. There are 40 different patterns which are chosen at random, so the game is full of endless variety. This is of particular value in helping young children to learn the art of concentrating and, at the same time, to introduce them to the computer.

METEOR AND TORPEDO ALLEY (L2/16K)
\$10.95 + 60c p&p

Those who frequent games arcades will recognize these two electronic games. In METEOR you must destroy the enemy space ships before they see you. In its most difficult mode, the odds are a thumping 238 to 1 against you being successful. In torpedo alley you must sink the enemy ships without hitting your own supply ship. Both games include sound effects and are remarkably accurate reproductions of the arcade games.

AUSTRALIAN SOFTWARE (Cont.)**GAMES****SHEEPDOG (L2/16K)****\$8.95 + 60c p&p**

Ever wondered how a sheepdog manages to drive all those awkward sheep into a pen? Well, here is your chance to find out just how difficult it is and have a lot of fun at the same time. You control the sheepdog, the computer controls the sheep! As if that isn't enough, look out for the dingoes lurking in the bush!

U BOAT**\$8.95 + 60c p&p**

Real time simulation at its best! Comes with working sonar-screen and periscope, a full rack of torpedoes, plenty of targets, working fuel and battery meters, helpful Mothership for high-seas reprovisioning and even has emergency radio for that terrible moment when the depth charges put your crew at risk. Requires Level II/16K.

SPACE INVADERS WITH SOUND**\$8.95 + 60c p&p**

Much improved version of this arcade favourite with redesigned laser and cannon blasts, high-speed cannon, 50 roving drone targets, 10 motherships and heaps of fun for all. Level II with 4K and 16K versions on this cassette.

GOLF (L2/16K)**\$8.95 + 60c p&p**

Pit your skills of mini-golf against the computer. Choose the level of difficulty, the number of holes and whether you want to play straight mini golf or crazy golf. Complete with hazards, water traps, bunkers and trees. Great fun for kids of all ages.

DOMINOES(L2/16K)**\$8.95 + 60c p&p**

Pit your skill at dominoes against the computer, which provides a tireless opponent. Another application of supergraphics from the stable of Charlie Bartlett. Dominoes are shown approximately life size in full detail (except for colour!). The monitor screen is a window which you can move from one end of the string of dominoes to the other. Best of all, you don't lose any pieces between games!

KID'S STUFF (formerly MMM-1)**\$8.95 + 60c p&p**

Three games on one cassette from that master of TRS-80 graphics, Charlie Bartlett. Includes INDY 500, an exciting road race that gets faster and faster the longer you play, SUBHUNT in which your warship blows up unfortunate little submarines all over the place, and KNEVEL (as in motorcycle, ramp and buses).

OTHER PROGRAMS**INFINITE BASIC BY RACET (32K/1 DISK)****\$49.95 + 50c. p&p**

Full matrix functions — 30 BASIC commands; 50 more STRING functions as BASIC commands.

GSF/L2/48K**\$24.95 + 50c. p&p**

18 machine language routines including RACET sorts.

BUSINESS ADDRESS AND INFORMATION SYSTEM (48K/DISK)**\$24.95 + 50c. p&p**

Allows you to store addresses and information about businesses, edit them and print them out.

HISPED (L216, 32 or 48K) \$29.95

This machine language program allows you to SAVE and LOAD programs and data to tape at speeds up to 2000 baud (4 times normal) using a standard cassette recorder. A switch must be installed to remove the XRX III loading board, if fitted.

LOWER CASE FOR YOUR TRS-80/SYSTEM 80**Kit only \$49.00 plus \$2.00 p&p**

Give your TRS-80 or SYSTEM 80 a lower case display with proper descenders and a block cursor (similar to the TRS-80 Model III). Also includes symbols for the four suits of cards. Includes full fitting instructions, all necessary components and a special machine language driver program to enable lower case in BASIC. The modification is similar to the Tandy model and does not work with Electric Pencil without further modifications.

These kits require disassembly of your computer and some soldering. They should only be installed by someone who has experience in soldering integrated circuits, using a low power, properly earthed soldering iron. If you do not have the necessary experience/equipment, we will install the modification for you for \$20 plus freight in both directions. Make sure you arrange the installation with us first, before despatching your computer, so that we can assure you of a rapid turn-around. We are also arranging to have installers in each State. See elsewhere in this issue for their names and addresses.

PRICES

Cat No.

HD-020 Lower case mod kit for TRS-80

\$49.00 plus \$2.00 p&p

HD-021 Lower case mod kit for SYSTEM-80

\$49.00 plus \$2.00 p&p**EPSON MX-80 PRINTER****ONLY \$949 Inc. Cable for TRS-80 and p&p****(*Printer only — \$940 incl. p&p)**

The EPSON MX-80 printer is compact, quiet, has features unheard of only 2-3 years ago in a printer at any price and, above all, is ultra-reliable. All available print modes may be selected under software control. Features include:

- high quality 9x9 dot-matrix character formation
- 3 character densities
 - . 80 characters per line at 10 chars/inch
 - . 132 characters per line at 16.5 chars/inch
 - . 40 characters per line at 5 chars/inch
- 2 line spacings
 - . 6 lines per inch 8 lines per inch
- 80 characters per second print speed
- bi-directional printing
- logical seeking of shortest path for printing
- lower case with descenders
- TRS-80 graphics characters built in
- standard Centronics printer port

The bi-directional printing coupled with the logical seeking of the shortest print path (which means that the print head will commence printing the next line from the end which requires the least travel, thereby minimising unutilised time) gives this printer a much higher throughput rate than many other printers quoting print speeds of 120 c.p.s. or even higher.

GREEN SCREEN SIMULATOR**\$9.50 incl. p&p**

The GREEN SCREEN SIMULATOR is made from a deep green perspex, cut to fit your monitor. It improves contrast and is much more restful to the eyes than the normal grey and white image.

All editorial staff of MICRO-80 are now using GREEN SCREEN SIMULATORS on their own monitors.

Please make sure to specify whether you have an old (suarish) or new (rounded) style monitor when ordering. Not available for Dick Smith monitors.

```

10 ' *****
    *   T. R. JONES   *
    *   HASTIE ST.   *
    *   TATURA       *
    *   3616         *
20 ' *****

30 CLS:DEFINTI-N:DIMX(100),Y(100):F$="#####.####"
40 PRINTTAB(19)"LINEAR REGRESSION ANALYSIS":PRINTTAB(19)
   STRING$(26,45)
50 PRINT:PRINT:PRINT:PRINTTAB(25)"LINEAR MODEL":PRINT:
   PRINTTAB(26)"Y = A + BX"
60 PRINT@977,"TYPE <ENTER> TO CONTINUE ....":INPUTI$
70 CLS:PRINTTAB(26)"ENTER DATA":PRINT:PRINT:
   PRINTTAB(13)"NUMBER OF DATA POINTS - MAXIMUM IS 100"
80 PRINTTAB(28)"*":INPUTND:IFND<20RND>100
   PRINTCHR$(27);CHR$(30);CHR$(29);CHR$(27):GOTO80
90 PRINT:PRINT:FORI=1TOND:PRINT"X(";I;") =":INPUTX(I):
   PRINTTAB(32)CHR$(27);"Y(";I;") =":INPUTY(I):NEXTI
100 CLS:K=0:PRINTTAB(25)"DATA ENTERED":PRINT:FORI=1TOND:K=K+1:
   PRINT"X(";I;") =":X(I);TAB(32)"Y(";I;") =":Y(I):
   IFK<10NEXTIELSEPRINT@977,"TYPE <ENTER> TO CONTINUE ....":
   INPUTI$:K=0:CLS:NEXTI
110 PRINT@983,"EDIT DATA (Y/N)":
120 I$=INKEY$:IFI$=""THEN120ELSEIFI$="Y"THEN130ELSEIFI$="N"
   THEN140ELSE120
130 CLS:K=0:INPUT"DATA POINT TO BE EDITED":I:
   IFI>NDTHEN130ELSEPRINTTAB(4)"X(";I;") =":X(I);TAB(36)"Y(";I;
) =":Y(I):
   PRINT"NEW X(";I;") =":INPUTX(I):
   PRINTTAB(32)CHR$(27);"NEW Y(";I;") =":INPUTY(I):GOTO100
140 CLS:PRINT@473,"COMPUTING ...."
150 X1=0:X2=0:Y1=0:Y2=0:XY=0:FORI=1TOND:X1=X1+X(I):
   X2=X2+X(I)[2:Y1=Y1+Y(I):Y2=Y2+Y(I)[2:
   XY=XY+(X(I)*Y(I)):NEXTI
160 X3=X1/ND:Y3=Y1/ND:R1=XY-(X1*Y1/ND):R2=X2-(X1[2/ND):
   R3=Y2-(Y1[2/ND):R=(R1[2)/(R2*R3):IFR>1THENR=1
170 R4=SQR(R):IFR1<0THENR4=-R4
180 B=R1/R2:A=Y3-(B*X3)
190 CLS:PRINT"THE EQUATION FOR THE REGRESSION LINE IS ":PRINT
200 IFB<0PRINT"YHAT =":A;"-":ABS(B);"X"
   ELSEPRINT"YHAT =":A;"+":B;"X"
210 PRINT:PRINT"MEAN X =":X3:PRINT"MEAN Y =":Y3
220 PRINT:PRINT"COEFFICIENT OF LINEAR CORRELATION =":R4
230 PRINT@978,"PRINT RESIDUAL TABLE (Y/N)":
240 I$=INKEY$:IFI$=""THEN240ELSEIFI$="Y"THEN260ELSEIFI$="N"
   THEN230ELSE240
250 CLS:J=0
260 FORI=1TOND:YH=A+(B*X(I)):RE=YH-Y(I)
270 IFJ=0GOSUB380
280 PRINTUSING"##";I;PRINTUSINGP$;X(I);Y(I);YH;RE
290 J=J+1:IFJ=>10THEN300ELSE310
300 PRINT@977,"TYPE <ENTER> TO CONTINUE ...":INPUTI$:J=0:CLS
310 NEXTI
320 SD=(1-R)*R3:PRINT:PRINT
   "SUM OF SQUARES OF DEVIATION FROM REGRESSION =":SD
330 PRINT@978,"ESTIMATE VALUES OF Y (Y/N)":
340 I$=INKEY$:IFI$=""THEN340ELSEIFI$="Y"THEN350ELSEIFI$="N"
   THEN370ELSE340
350 CLS
360 INPUT"X =":XX:YY=A+(B*XX):PRINTTAB(32)CHR$(27);
   "Y =":YY:GOTO360
370 CLS:END
380 CLS:PRINT"POINT":TAB(11)"X":TAB(25)"Y":TAB(38)"YHAT":
   TAB(52)"RES":PRINT:RETURN

```

***** BIG LETTERS LII/16K *****
 (C) Ronald J. Sully

Have you ever taken a close look at your VDU and noticed all those little dots that make up each character. Now you can see those dots as large as life, but instead of a dot, each element is made up of two graphic blocks to form one symmetrical square representing a dot. BIG LETTERS allows you to draw the full character set of your '80 larger than life from ASCII code 32 (space) to code 90 (Z) by simply pressing the corresponding key. This program has a number of obvious uses such as displaying messages which can be read by a number of people at the same time.

It could also form the basis of educational programs for small children - computerised flash cards, for example. We would welcome submissions of such programs from our readers. Each of the big letters occupies 21 print spaces (7 spaces wide by 3 lines high), thereby allowing 9 letters per line. Apart from the letters there are 9 control keys:-

KEY	ASCII CODE	FUNCTION
shift A	97	Abort message. This function clears the screen. Returns the cursor to 0,0 and cancels any message you may have compiled.
shift C	99	If the cursor is on, it turns it off. If the cursor is off, it turns it on.
shift N	110	Changes the characters from normal to negative and from negative to normal.
CLEAR (System '80) (use CTRL)	31 10	This function clears the screen and returns the cursor to 0,0 and forms part of your message.
ENTER/NEWLINE	13	Causes a linefeed and puts the cursor at the beginning of the next line. This function also forms part of the message and counts as one character. Do not try to backspace over this function.
LEFT ARROW (BACKSPACE)	8	Backspaces and deletes the last character. Used to correct all those mistakes.
shift M	109	Change modes from compiling message to writing message or vice versa. NOTE: this is the only function available to you during the printing mode.
shift L	108	Allows you to load a previously recorded message.
shift S	115	Allows you to save a message to tape. NOTE: if your message contains punctuation that performs special functions in BASIC (e.g. ",") you will not be able to load all of your message back in.

The program allows a limit of 250 characters, including control characters, in your message. Automatic screen scrolling occurs when the screen is full. If you wish to fill the screen without it scrolling when the message is printed, press the CLEAR key "after it has scrolled" in the writing stage. Now! Let's get the program typed in. You will have noticed by now that it is not exactly short and that there is an awful lot of data. WELL! You should subscribe to the cassette edition!

Key in the program very carefully, noting that the string arrays N\$(1),NE\$(1) to N\$(59) contain 22 spaces, (they only need 21 but enter 22 to be sure). I suggest that you make the ON ERROR GOTO statement in line 30 a REMark until you are sure that you have typed in the program correctly. Errors will normally occur if you press the wrong keys (up arrow, esc, etc.) and an FC error will also occur while the error trap is disabled, so ignore these. Line 1100 is NOT a mistake. It should be a REMark (but not for long). After you have finished typing in the data and provided that your fingers still work, check the program for typing mistakes. If you think that there are no mistakes then CSAVE it and CLOAD? it. Now comes the fun part. Type RUN and press ENTER/NEWLINE. The screen will clear, there will be a short pause then the message "INITIALIZING" will appear on the screen, rapidly followed by each of the ASCII character set (32 to 90) in succession, in 32 characters per line format, (sorry System '80 users). It is at this stage that all those spaces in N\$ and NE\$ are being converted into graphic codes. After this the screen will clear again and the message EDIT 1100 will appear and the program will place you in the edit mode for line 1100. Now press the "D" key and then press the ENTER/NEWLINE key; you just deleted the REMark from line 1100 and turned it into the basic statement RETURN. Run the program again and check out all the characters in turn. If everything is O.K. you can now remove the REMark statement from line 30 to activate the error trap and you can also delete lines 1110 to 1750. YES that's right! All that data that took you hours to type in is no longer required in the program. GO ON! Type DELETE 1110-1750 and press ENTER/NEWLINE. Now you can CSAVE this version knowing that it will not take as long to CLOAD when you need it again. Don't be surprised by those funny looking strings; they are only SUPER GRAPHICS (see Issue 5, page 16). All that is left for you to do now is run the program and try your hand at advertising.

```

10 ' *****
    * RONALD J. SULLY *
    * PACKHAM PLACE *
    *   CANBERRA   *
    *   ACT 2615   *
    * *****
20 '

30 CLS: CLEAR 1000: DEFINT A-Z: DIM N$(60), NE$(60), S1(60), S2(60): ONERR
  OR GOTO 320
40 CL$=CHR$(199): GOSUB 510
50 M$="": MO$="": MO=100
60 P=-7
70 GOSUB 260
80 CU=P+129: IFC0=0 THEN PRINT@CU, CHR$(140);
90 GOSUB 300
100 IFL=200 THEN CLS: M$=M$+CHR$(L): MO$=MO$+CHR$(MO): GOTO 60
110 IFL=8 THEN M$=LEFT$(M$, LEN(M$)-1): MO$=LEFT$(MO$, LEN(MO$)-1): GOSUB 240: GOTO 80
120 IFL=100 THEN M$=M$+CHR$(L): MO$=MO$+CHR$(MO): GOSUB 290: PRINT@CU, CL$: GOSUB 270: GOTO 80
130 IFL=110 THEN IF MO=100 THEN MO=101: GOTO 90 ELSE MO=100: GOTO 90
140 IFL=99 THEN IFC0=0: PRINT@CU, CL$: C0=1: GOTO 90 ELSE C0=0: GOTO 80
150 IFL=109 GOTO 330
160 IFL=97 THEN CLS: GOTO 50
170 IFL=108 OR L=115 THEN 460
180 M$=M$+CHR$(L): MO$=MO$+CHR$(MO)
190 FOR C=0 TO 2: K=C*7+1: PRINT@P+64*C, " ";
200 ON MO-99 GOTO 210, 220
210 PRINT MID$(N$(L-31), K, 7);: NEXT: GOTO 70
220 PRINT MID$(NE$(L-31), K, 7);: NEXT: GOTO 70
230 ' ROUTINE TO BACKSPACE
240 IF P/64=INT(P/64) THEN P=P-136 ELSE P=P-7
250 PRINT@CU, CL$: FOR B=PTOP+128 STEP 64: PRINT@B, CL$: NEXT: RETURN
260 IF P<>0 AND (P-56)/64=INT((P-56)/64) THEN P=P+136 ELSE P=P+7
270 IF P=960 THEN FORT=1 TO 4: PRINT: NEXT: P=768
280 RETURN
290 P=INT(P/64)*64+192: RETURN
300 I$=INKEY$: IF I$="" THEN 300 ELSE L=ASC(I$): IFL=13 THEN NL=100 ELSE IF L=31 THEN NL=200
  SYSTEM 80 USERS WILL NEED TO DELETE .. IFL=31.. AND INSERT
  .. IFL=10..

310 RETURN
320 IF ERL=380 THEN RESUME NEXT ELSE RESUME 80
330 CLS: IF M$="" THEN PRINT@960, "NO MESSAGE";: FORT=1 TO 2000: NEXT: CLS: GOTO 50
340 P=0: FOR K=1 TO LEN(M$): CH=ASC(MID$(M$, K, 1)): MO=ASC(MID$(MO$, K, 1))
350 IF CH=100 THEN GOSUB 290: GOSUB 270: GOTO 390
360 IF CH=200 THEN FORT=1 TO 2000: NEXT: CLS: P=0: GOTO 390
370 GOSUB 410: GOSUB 450: IF I$="m" THEN CLS: GOTO 50
380 IF (K<LEN(M$) AND ASC(MID$(M$, K+1, 1))<>200) THEN GOSUB 260
390 NEXT K: FORT=1 TO 2500: NEXT: CLS: GOTO 330
400 ' PRINT CHARACTER IN SELECTED MODE
410 FOR C=0 TO 2: PRINT@P+64*C, " ";
420 IF MO=101 THEN PRINT MID$(NE$(CH-31), C*7+1, 7);: GOTO 440
430 PRINT MID$(N$(CH-31), C*7+1, 7);
440 NEXT: RETURN
450 I$=INKEY$: RETURN' RETURN TO START
460 CLS: PRINT CHR$(23): PRINT "PREPARE CASSETTE TO ";: IFL=108 THEN PRINT "LOAD...": ELSE PRINT "SAVE..."
470 PRINT
  PRESS ENTER WHEN READY....: I$=INKEY$
480 IF INKEY$="" THEN 480
490 IFL=108 THEN INPUT#-1, M$: INPUT#-1, MO$: GOTO 330
500 PRINT#-1, M$: PRINT#-1, MO$: CLS: GOTO 60
510 N$(1)="": NE$(1)="
"
520 N$(2)="": NE$(2)="
"
530 N$(3)="": NE$(3)="
"
540 N$(4)="": NE$(4)="
"
550 N$(5)="": NE$(5)="
"
560 N$(6)="": NE$(6)="
"
570 N$(7)="": NE$(7)="
"

```


580 N\$(8)="	" : NE\$(8) = "
"	
590 N\$(9)="	" : NE\$(9) = "
"	
600 N\$(10)="	" : NE\$(10) = "
"	
610 N\$(11)="	" : NE\$(11) = "
"	
620 N\$(12)="	" : NE\$(12) = "
"	
630 N\$(13)="	" : NE\$(13) = "
"	
640 N\$(14)="	" : NE\$(14) = "
"	
650 N\$(15)="	" : NE\$(15) = "
"	
660 N\$(16)="	" : NE\$(16) = "
"	
670 N\$(17)="	" : NE\$(17) = "
"	
680 N\$(18)="	" : NE\$(18) = "
"	
690 N\$(19)="	" : NE\$(19) = "
"	
700 N\$(20)="	" : NE\$(20) = "
"	
710 N\$(21)="	" : NE\$(21) = "
"	
720 N\$(22)="	" : NE\$(22) = "
"	
730 N\$(23)="	" : NE\$(23) = "
"	
740 N\$(24)="	" : NE\$(24) = "
"	
750 N\$(25)="	" : NE\$(25) = "
"	
760 N\$(26)="	" : NE\$(26) = "
"	
770 N\$(27)="	" : NE\$(27) = "
"	
780 N\$(28)="	" : NE\$(28) = "
"	
790 N\$(29)="	" : NE\$(29) = "
"	
800 N\$(30)="	" : NE\$(30) = "
"	
810 N\$(31)="	" : NE\$(31) = "
"	
820 N\$(32)="	" : NE\$(32) = "
"	
830 N\$(33)="	" : NE\$(33) = "
"	
840 N\$(34)="	" : NE\$(34) = "
"	
850 N\$(35)="	" : NE\$(35) = "
"	
860 N\$(36)="	" : NE\$(36) = "
"	
870 N\$(37)="	" : NE\$(37) = "
"	
880 N\$(38)="	" : NE\$(38) = "
"	
890 N\$(39)="	" : NE\$(39) = "
"	
900 N\$(40)="	" : NE\$(40) = "
"	
910 N\$(41)="	" : NE\$(41) = "
"	
920 N\$(42)="	" : NE\$(42) = "
"	
930 N\$(43)="	" : NE\$(43) = "
"	
940 N\$(44)="	" : NE\$(44) = "
"	
950 N\$(45)="	" : NE\$(45) = "
"	
960 N\$(46)="	" : NE\$(46) = "
"	
970 N\$(47)="	" : NE\$(47) = "
"	

```

980 N$(48)="          ":"NE$(48)="
"
990 N$(49)="          ":"NE$(49)="
"
1000 N$(50)="          ":"NE$(50)="
"
1010 N$(51)="          ":"NE$(51)="
"
1020 N$(52)="          ":"NE$(52)="
"
1030 N$(53)="          ":"NE$(53)="
"
1040 N$(54)="          ":"NE$(54)="
"
1050 N$(55)="          ":"NE$(55)="
"
1060 N$(56)="          ":"NE$(56)="
"
1070 N$(57)="          ":"NE$(57)="
"
1080 N$(58)="          ":"NE$(58)="
"
1090 N$(59)="          ":"NE$(59)="
"
1100 'RETURN
1110 FOR T=1 TO 59: S1(T)=PEEK(VARPTR(N$(T))+2)*256+PEEK(VARPTR(N$(T)+1)): S2(T)=PEEK(VARPTR(NE$(T))+2)*256+PEEK(VARPTR(NE$(T)+1)): NEXT
1120 PRINT@20,CHR$(23);"BIG LETTERS":PRINT@260,"INITIALIZING - -
- - -";
1130 FOR T=1 TO 59:PRINT@538,CHR$(T+31);:FORM=0 TO 20:READ G:POKE S1(T)+M,G:POKE S2(T)+M,319-G:NEXT M,T
1140 CLS:PRINT@192,"EDIT 1100":EDIT 1100
1150 I$=INKEY$:IFI$="" THEN 1150 ELSE L=ASC(I$):RETURN
1160 '

```

GRAPHIC CODES FOR CHARACTERS

```

1170 DATA128,128,128,128,128,128,128,128,128,128,128,128,128,128
,128,128,128,128,128,128,128:' SPACE
1180 DATA128,128,128,188,128,128,128,128,128,128,191,128,128,128
,128,128,128,140,128,128,128:' !
1190 DATA128,128,188,128,188,128,128,128,128,128,131,128,131,128,128
,128,128,128,128,128,128,128:' "
1200 DATA128,128,188,128,188,128,128,128,179,191,179,191,179,128
,128,128,143,128,143,128,128:' #
1210 DATA128,128,176,188,176,176,128,128,131,140,191,140,176,128
,128,131,131,143,131,128,128:' $
1220 DATA128,188,188,128,128,176,128,128,128,176,140,131,128,128
,128,131,128,128,143,143,128:' %
1230 DATA128,176,140,176,128,128,128,128,179,140,179,128,176,128
,128,131,140,140,131,140,128:' &
1240 DATA128,128,188,188,128,128,128,128,128,140,131,128,128,128
,128,128,128,128,128,128,128:' '
1250 DATA128,128,128,176,140,128,128,128,128,191,128,128,128,128
,128,128,128,131,140,128,128:' (
1260 DATA128,128,140,176,128,128,128,128,128,128,191,128,128
,128,128,140,131,128,128,128:' )
1270 DATA128,176,128,188,128,176,128,128,140,191,191,191,140,128
,128,131,128,143,128,131,128:' *
1280 DATA128,128,128,176,128,128,128,128,140,140,191,140,140,128
,128,128,128,131,128,128,128:' +
1290 DATA128,128,128,128,128,128,128,128,128,128,188,188,128,128,128
,128,140,131,128,128,128,128:' ,
1300 DATA128,128,128,128,128,128,128,128,140,140,140,140,140,128
,128,128,128,128,128,128,128:' -
1310 DATA128,128,128,128,128,128,128,128,128,128,128,128,128,128
,128,128,143,143,128,128,128:' .
1320 DATA128,128,128,128,128,176,128,128,128,176,140,131,128,128
,128,131,128,128,128,128,128:' /
1330 DATA128,176,140,140,140,176,128,128,191,176,140,131,191,128
,128,131,140,140,140,131,128:' 0
1340 DATA128,128,176,188,128,128,128,128,128,128,191,128,128,128
,128,128,140,143,140,128,128:' 1
1350 DATA128,176,140,140,140,176,128,128,176,140,140,140,143,128
,128,143,140,140,140,140,128:' 2
1360 DATA128,176,140,140,140,176,128,128,128,128,140,140,179,128
,128,131,140,140,140,131,128:' 3

```

1370 DATA128,128,128,176,188,128,128,128,188,179,176,191,176,128
128,128,128,128,143,128,128: ' 4
1380 DATA128,188,140,140,128,128,128,131,131,131,131,188,128
128,131,140,140,140,131,128: ' 5
1390 DATA128,128,176,140,140,128,128,128,191,140,140,140,176,128
128,131,140,140,140,131,128: ' 6
1400 DATA128,140,140,140,140,188,128,128,128,176,140,131,128,128
128,143,128,128,128,128,128: ' 7
1410 DATA128,176,140,140,140,176,128,128,179,140,140,140,179,128
128,131,140,140,140,131,128: ' 8
1420 DATA128,176,140,140,140,176,128,128,131,140,140,140,191,128
128,128,140,140,131,128,128: ' 9
1430 DATA128,128,188,188,128,128,128,128,128,128,128,128,128,128
128,128,143,143,128,128,128: ' :
1440 DATA128,128,188,188,128,128,128,128,128,188,188,128,128,128
128,140,131,128,128,128,128: ' ;
1450 DATA128,128,128,176,140,128,128,128,128,140,179,128,128,128,128
128,128,128,131,140,128,128: ' <
1460 DATA128,128,128,128,128,128,128,128,179,179,179,179,179,128
128,128,128,128,128,128,128: ' =
1470 DATA128,128,140,176,128,128,128,128,128,128,128,179,140,128
128,128,140,131,128,128,128: ' >
1480 DATA128,176,140,140,140,176,128,128,128,128,176,140,131,128
128,128,128,140,128,128,128: ' ?
1490 DATA128,176,140,140,140,176,128,128,176,140,188,128,191,128
128,131,140,143,140,131,128: ' @
1500 DATA128,128,176,140,176,128,128,128,128,191,176,176,176,191,128
128,143,128,128,128,143,128: ' A
1510 DATA128,140,188,140,140,176,128,128,128,191,140,140,179,128
128,140,143,140,140,131,128: ' B
1520 DATA128,176,140,140,140,176,128,128,191,128,128,128,128,128
128,131,140,140,140,131,128: ' C
1530 DATA128,140,188,140,140,176,128,128,128,191,128,128,191,128
128,140,143,140,140,131,128: ' D
1540 DATA128,188,140,140,140,140,128,128,191,140,140,128,128,128
128,143,140,140,140,140,128: ' E
1550 DATA128,188,140,140,140,140,128,128,191,140,140,128,128,128
128,143,128,128,128,128,128: ' F
1560 DATA128,176,140,140,140,140,128,128,191,128,128,140,188,128
128,131,140,140,140,143,128: ' G
1570 DATA128,188,128,128,128,188,128,128,191,140,140,140,191,128
128,143,128,128,128,143,128: ' H
1580 DATA128,128,140,188,140,128,128,128,128,191,128,128,128
128,128,140,143,140,128,128: ' I
1590 DATA128,128,128,128,128,188,128,128,128,128,128,128,191,128
128,131,140,140,140,131,128: ' J
1600 DATA128,188,128,128,176,140,128,128,191,140,179,128,128,128
128,143,128,128,131,140,128: ' K
1610 DATA128,188,128,128,128,128,128,128,191,128,128,128,128,128
128,143,140,140,140,140,128: ' L
1620 DATA128,188,176,128,176,188,128,128,191,128,143,128,191,128
128,143,128,128,128,143,128: ' M
1630 DATA128,188,176,128,128,188,128,128,191,128,131,140,191,128
128,143,128,128,128,143,128: ' N
1640 DATA128,176,140,140,140,176,128,128,191,128,128,128,191,128
128,131,140,140,140,131,128: ' O
1650 DATA128,188,140,140,140,176,128,128,191,140,140,140,131,128
128,143,128,128,128,128,128: ' P
1660 DATA128,176,140,140,140,176,128,128,191,128,176,128,191,128
128,131,140,140,131,140,128: ' Q
1670 DATA128,188,140,140,140,176,128,128,191,140,188,140,131,128
128,143,128,128,128,131,140,128: ' R
1680 DATA128,176,140,140,140,176,128,128,131,140,140,140,176,128
128,131,140,140,140,131,128: ' S
1690 DATA128,140,140,188,140,140,128,128,128,191,128,128,128
128,128,128,143,128,128,128: ' T
1700 DATA128,188,128,128,188,128,128,191,128,128,128,191,128
128,131,140,140,140,131,128: ' U
1710 DATA128,188,128,128,128,188,128,128,131,188,128,188,131,128
128,128,128,143,128,128,128: ' V
1720 DATA128,188,128,128,128,188,128,128,191,128,176,128,191,128
128,143,131,128,131,143,128: ' W
1730 DATA128,188,128,128,128,188,128,128,128,179,140,179,128,128
128,143,128,128,128,143,128: ' X
1740 DATA128,188,128,128,128,188,128,128,131,188,131,128,128
128,128,128,143,128,128,128: ' Y
1750 DATA128,140,140,140,140,188,128,128,176,140,131,128,128
128,143,140,140,140,140,128: ' Z

***** COSMIC WARS 32K/DISK *****
(C) P. Holliday

This program simulates an arcade game called "COSMIC KNOCKOUT". The program has sound effects which can be heard through the cassette deck, (System 80 owners should type OUT254,255 before running the program so as to hear the sound through their external tape recorder). Use the following keys to move:-

TRS-80	SYSTEM '80	ACTION
UP ARROW	ESC	MOVE SIGHTS UP (enemy goes down)
DOWN ARROW	CTRL	MOVE SIGHTS DOWN (enemy goes up)
LEFT ARROW	<	MOVE SIGHTS LEFT (enemy goes right)
RIGHT ARROW	>	MOVE SIGHTS RIGHT (enemy goes left)
SPACE BAR	SPACE BAR	FIRE !!!

Before each game starts you are asked for a rating between 1 and 9; 9 is slow, 1 is fast. You have a time limit of 90 seconds. The timer is displayed on the screen throughout the game along with your score. You will be rated on your score after the completion of each game.

```

10 ' *****
    *   PETER HOLLIDAY   *
    *   BANKSIDE ST.    *
    *   NATHAN           *
    *   BRISBANE 4111    *
    * *****
20 '

30 'WRITTEN FOR A 32K (MIN) DISK SYSTEM, RUN
    ONLY UNDER DISK BASIC.
40 CLEAR1000:CLS:PRINT@512+25,"PLEASE WAIT";
50 DEFUSR1=&H7D00
60 DEFUSR2=27100
70 DATA120,170,190,129,81,136,140,151,190,129
80 FORI=1TO5:READK(I),KK(I):NEXTI
90 DATA33,211,63,54,190,33,212,63,54,129,33,231,63,54,130,33,232
    ,63,54,189,33,148,63,54,184,33,149,63,54,135,33,166,63,54,139,33,
    167,63,54,180,33,85,63,54,160,33,86,63,54,158,35,54,129,33,100,63
    ,54,130,35,54
100 DATA173,35,54,144,33,23,63,54,184,35,54,135,33,35,63,54,139,
    35,54,180,33,216,62,54,160,35,54,158,35,54,129,33,225,62,54,130,3
    5,54,173,35,54,144,33,154,62,54,184,35,54,135,33,160,62,54,139,35
    ,54,180,33,91
110 DATA62,54,160,35,54,158,35,54,129,33,94,62,54,130,35,54,173,
    35,54,144,33,29,62,54,176,35,54,176,201,0
120 I=0
130 READN:IFN=0THEN150
140 POKE32000+I,N:I=I+1:GOTO130
150 DATA62,16,211,254,221,33,18,106,221,78,0,121,183,200,221,70,
    1,62,5,211,255,16,254,221,70,1,62,6,211,255,16,254,13,194,234,105
    ,221,35,221,35,1,255,255,33,48,0,9,218,10,106,195,228,105
160 FORI=27100TO27152:READY:POKEI,Y:NEXT
170 NN=27152
180 DIMN(10),M(10)
190 GOTO200:DATA86,120,91,100,96,102,75,109,60,114,50,121,40,129
    ,30,137,20,145,10,152,10
200 FORI=1TO10:READN(I),M(I)
210 IFI=1ANDN(I)<>86THEN200
220 NEXTI
230 FORP=1TO1000:NEXT:CLS:PRINT@10,"- C O S M I C W A R S - B
    O -";:FORH=1TO4:GOSUB680:NEXTH:FORI=30TO1STEP-1:POKENN+2,255:POKE
    NN+3,20:POKENN+4,00:K=USR2(0):PRINT@74+I,"=";:FORK=1TOI:NEXTK:NEX
    TI
240 T=0:L=0
250 CLS:LINEINPUT"WHAT IS YOUR RATING (1-9) ? ":V$:V=VAL(V$)
260 IFV<1ORV>9THEN250
270 DEFSTRA-D:A=CHR$(191):A2=CHR$(188):A1=CHR$(143):B=STRING$(5,
    140)
280 C=CHR$(157)+CHR$(183)+CHR$(187)+CHR$(174)
290 F=0:S=0
300 CLS:PRINT@30,A;:PRINT@30+64,A1;:PRINT@512,B;:PRINT@570,B;:PR
    INT@960+30,A;:PRINT@960-34,A2;

```



```

310 T=T+.005:IFT>.90THENGOTO610
320 PRINT@542,"+";
330 PRINT@960,"SCORE -"L;
340 PRINT@0,"TIME -"USING"##.##";T;
350 IFZZ=-1THEN360ELSEPRINT@X,C;
360 ONERRORGOTO590
370 K=15359
380 IFF=1THEN400
390 X=RND(64):X=16*X+X:PRINT@X,C;:F=1
400 Z=RND(4):IFZ=1THENX=X-1
410 IFZ=2THENX=X+1
420 IFZ=3THENX=X+64
430 IFZ=4THENX=X-64
440 IFZZ=-1THENG=X
450 ZZ=0
460 IFPEEK(K)=16THENX=X-63+Z
470 IFPEEK(K)=8THENX=X+65-Z
480 IFPEEK(K)=64THENX=X-2
490 IFPEEK(K)=32THENX=X+2
500 IFPEEK(K)=128THEN540
510 FORI=1TO(V*2):NEXT
520 G=X
530 GOTO300
540 PF=USR1(0)
550 POKENN+2,255:POKENN+3,33:POKENN+4,00:M=USR2(0)
560 IF535<GAND543>6THEN570ELSE580
570 FORJ=1TO2:PRINT@538,"ZAP!";:GOSUB680:PRINT@541,STRING$(4,191);:GOSUB680:NEXTJ:PRINT@541,"    ";:L=L+1:ZZ=-1:GOTO290
580 FORI=1TO10:NEXT:GOTO300
590 IFX>1020THENX=X-960:RESUMEELSEX=X+960:RESUME
600 PRINTPEEK(15359);:GOTO600
610 FORI=1TO5:POKENN+2,K(I):POKENN+3,K(I):POKENN+4,00:V=USR2(0):NEXT
620 FORI=1TO500:NEXT
630 PRINT@512,"YOUR SCORE WAS "L". RATING";:
640 IFL<3THENPRINT"- GRANNY!!";:FORI=1TO1000:NEXT:GOTO230
650 IFL<5THENPRINT"- ROOKIE!!";:FORI=1TO1000:NEXT:GOTO230
660 PRINT"-- ACE!!!!!!";:FORU=1TO6:GOSUB680:NEXT:GOTO230
670 END
680 FORI=10TO1STEP-1:POKENN+2,N(I):POKENN+3,M(I)/3:POKENN+4,00:M=USR2(0):NEXT
690 RETURN
700 END

```

***** UTIL 1 : UTIL 2 : UTIL 3 : UTIL 4 m/l *****
 (C) B.J. White

These four machine language programs are almost as remarkable for the way in which the author wrote them as for their intrinsic merit. The programs were actually hand-assembled without the use of an EDITOR/ASSEMBLER. Since MICRO-80 prefers a source listing for publication, the author then loaded in a disassembler and printed out a disassembled listing. A disassembled listing, of course, has no REMARKS so, he then typed in the remarks for UTIL 1. The other listings are not so comprehensibly REMARKed but UTIL 1 source code gives the idea. So what do they do?

UTIL 1 and UTIL 2 are identical in function, providing a flashing cursor, auto repeat of characters and a variable scroll routine. They only differ in the memory locations used, UTIL 1 being for low memory, (memory size not required) and UTIL 2 being for the top of 4K. Note that in UTIL 1 the initialisation routine is at the high memory end of the program whilst in UTIL 2, it is at the low memory end. As each program initialises only once, this makes maximum use of the available memory, as the initialisation routine is overwritten by BASIC in UTIL 1, although this may be prevented by changing location 4392 from 85 to AA. With UTIL 2 the initialisation routine may be saved by setting memory size. Set the memory size to 20308 to preserve the initialisation routine or set it to 20323 to gain more memory. UTIL 2 may be easily relocated; just change ALL 4F bytes to suit the locations used, e.g. change to 7F for the top of 16K.

ENTERING THE PROGRAM

You will need either a monitor program such as BMON (MICRO-80 issues 3,4 & 5) or an editor/assembler. If using a monitor, start editing memory at the address shown at the top of the left hand column (e.g. 42E9 Hex for UTIL 1) and change each memory address to the value shown in the next column. E.g. for UTIL 1, change 42E9 to F5; 42EA to 21, 42EB to 84; 42EC to 43, 42ED to 79, etc. If you are using an editor/assembler, enter the source code from column 3 across, then assemble in the normal manner.

HOW TO USE UTIL 1 OR UTIL 2.

By entering PRINT CHR\$(1) manual control of the scrolling is obtained. Once the screen is full, a loop is entered. If you want to scroll up one line press the up arrow (ESC). Pressing CLEAR will clear the screen and fill the screen with the next page, while pressing BREAK will return you to the command mode. However, you will have to press the up arrow a few times to get the prompt onto the bottom line. To return to normal screen operation, type PRINT CHR\$(0).

The repeat character routine waits a short period when a key is pressed then, if it is still held down, it will keep repeating until that key is released or until the line is full.

UTIL 3 and UTIL 4 are the same as UTIL 1 and UTIL 2 except that they incorporate a lower-case driver for those who have carried out the lower-case mod as published in MICRO-80 Issue 5, April, 1980. Using one of these programs on a suitably modified computer try the following:-

```
10 CLS:FOR X = 123 TO 127 :PRINT X;CHR$(X);:NEXT:PRINT
20 FOR X = 192 TO 223 :PRINT X;CHR$(X);:NEXT
```

After running the above program type PRINT CHR\$(1) and RUN it again using the up arrow and clear keys.

Using a suitable monitor, type in your choice of programs using the addresses given below:-

PROGRAM	START	END	ENTRY (HEX)	ENTRY (DEC)	MEMORY SIZE
UTIL 1	42E9	43A9	4385	17285	N/A
UTIL 2	4F55	4FFF	4F55	20309	20308/20323
UTIL 3	42E9	43CE	43AA	17322	N/A
UTIL 4	4F30	4FFF	4F30	20272	20271/20286

		NAME UTIL1	
42E9:	F5	SCROLL	PUSH AF
42FA:	21 84 43		LD HL, 4384 POINT TO SCROLL ON/OFF FLAG
42ED:	79		LD A,C PUT THE CHAR. IN A
42EE:	E6 FE		AND FE LOW BIT OFF
42F0:	20 03		JR NZ, 42F5 IF NOT ZERO NOFLAG
42F2:	71		LD (HL), C ELSE STOW THE NEW FLAG
42F3:	18 2D		JR 4322 AND GO TO BYPASS
42F5:	7E	NOFLAG	LD A, (HL) GET THE FLAG
42F6:	B7		OR A TEST IF ON or OFF
42F7:	28 28		JR Z, 4324 IF OFF RST
42F9:	3A 22 40		LD A, (4022) GET THE CURSOR CHAR.
42FC:	B7		OR A TEST IF ON or OFF
42FD:	20 25		JR NZ, 4324 RST IF ON
42FF:	79		LD A,C CHAR. TO A
4300:	FE 0D		CF 0D NEW LINE ?
4302:	20 20		JR NZ, 4324 RST IF NOT
4304:	2A 20 40		LD HL, 4020 GET CURSOR ADDRESS
4307:	11 40 00		LD 0E, 0040 LENGTH
430A:	19		ADD HL, DE ADD THEM
430B:	7C		LD A,H NEW MSB
430C:	FE 40		CF 40 WILL THE SCREEN OVERFLOW ?
430F:	20 14		JR NZ, 4324 RST IF NOT
4310:	CD 49 00	KBIN	CALL 0049 CALL KEYBOARD ROM ROUTINE
4313:	FE 01		CF 01 BREAK KEY ?
			READY IF SO
4317:	FE 3B		CF 3B UP ARROW ?
4319:	28 09		JR Z, 4324 RST IF SO
431B:	FE 1F		CF 1F CLS ?
431D:	20 11		JR NZ, 4310 KBIN IF NOT
431F:	CD 09 01	CLS	CALL 0109 CALL ROM CLEAR SCREEN ROUTINE
4322:	11	BYPASS	POP AF RESTORE REGISTERS
4323:	C9		RET AND BYPASS ROM
4324:	F1	RST	POP AF RESTORE REGISTERS
			AND RETURN TO ROM
4328:	8E 0E	READY	LD A, 0E TURN CURSOR ON
432A:	32 22 10		LD 1022, A
432D:	C3 10 06		JP 06CC AND RETURN TO BASIC READY PROMPT
4330:	E5	REPEAT	PUSH HL
			CALL FLASH
4334:	21 00 03		LD HL, 0300 PAUSE BEFORE STARTING TO REPEAT
4337:	A7		AND A
4338:	28 05		OR Z, 433F TO NOT NEW

```

                                SAVE THE CHAR.
                                TO EXIT
4330: 18 21      JR 4360
433F: 21 01 38 NOT NEW LD HL,8800
                                IF KEY NOT FRESHLY PUSHED
4342: B6        OR (HL)
                                TEST IF ANY KEYS ARE DOWN ANYWAY
4343: 0B 25     SLA 1
                                KLOOP
4348: A7        AND A
4349: 28 15     JR Z,4360 IF NOT EXIT
                                REPEAT COUNTER TO HL
434E: 2B        DEC HL
434F: 7D        LD A,H
4350: A7        AND A
4351: 0E 00     LD A,00
4358: 20 08     JR NZ,4360 EXIT IF NOT ZERO
4355: 3F 03     LD A,03
                                ELSE RESET
                                THE COUNTER
435A: 21 A0 01 LD HL,01A0
                                PAUSE BETWEEN EACH CHAR REPEAT
                                FETCH THE CHAR AS IF IT WAS JUST PUSHED
                                ADJUST THE REPEAT COUNTER
4363: E1        POP HL
4364: C9        RET

4365: E5        FLASH PUSH HL
                                GET THE FLASH COUNT
4369: 2B        DEC HL
436A: 7D        LD A,H
436B: A7        AND A
436D: 20 0A     JR NZ,437B IF NOT ZERO FSKIP
436E: 2A 20 40 LD HL,(4020) GET CUROR POSITION
4371: 7E        LD A,(HL)
                                SHOW THE CHAR.
4372: EE 00     XOR 00
                                TOGGLE THE CHAR.
4374: 77        LD (HL),A
                                SAVE THE NEW CHAR.
                                DELAY FOR FLASH RATE
                                SAVE FLASH COUNT
4378: E1        POP HL
4379: C9 18 03 JR C,03E3 RETURN TO REPEAT ROUTINE VIA ROM
437F: 00        NOP
                                FLASH COUNT LSB
4380: 03        INC DE
                                " " MSB
4381: 00        POP DE
                                REPEAT COUNT LSB
4382: 03        INC DE
                                " " MSB
4383: 00        POP DE
                                CHAR TO BE REPEATED
4384: 00        NOP
                                SCROLL ON/OFF FLAG
4385: 21 19 42 INIT LD HL,42E9
                                POINT TO SCROLL
4386: 22 1F 40 LD (401E),HL
                                AND ADJUST DCB
                                POINT TO REPEAT
438I: 22 15 40 LD (4016),HL
                                AND ADJUST DCB
                                POINT TO END +1
                                and ADJUST POINTERS
4394: AF        XOR A
4395: 77        LD (HL),A
                                NOTE CHANGE MEMORY LOCATION
4396: 23        INC HL
                                4392 from 85 to AA
4397: 77        LD (HL),A
                                if you do NOT want the
4398: 22 A1 40 LD (40A1),HL
                                INIT. routine to be over-
439B: 23        INC HL
                                written.
439C: 77        LD (HL),A
439D: 23        INC HL
439F: 22 19 40 LD (40F2),HL
43A1: 22 FB 40 LD (40FB),HL
43A1: 22 FD 40 LD (40FD),HL
43A7: 03 0C 06 JIP 040C RETURN TO BASIC PROMPT

NAME UTIL2
4F63: 21 64      INIT LD HL,4F64
4F68: 22 1E 40 LD (401E),HL
4F6B: 21 AF      LD HL,AFAB
4F6F: 22 16 40 LD (4016),HL
4F61: 03 0C 06 JIP 040C RETURN TO BASIC
4F64: F5        SCROLL PUSH AF
4F65: 21 FF      LD HL,FFFF
4F68: 79        LD A,C
4F69: E6 FE      AND FE
4F6B: 20 03     JR NZ,4F70
4F6C: 71        LD (HL),0
4F6D: 18 2D     JR 4F9B
4F70: 7E        LD A,(HL)

```

4F71:	B7	OR	A
4F72:	28 28	JR	Z, 4F9F
4F74:	3A 22 40	LD	A, (4022)
4F77:	B7	OR	A
4F78:	20 25	JR	NZ, 4F9F
4F7A:	79	LD	A, C
4F7B:	FE 0D	CP	0D
4F7D:	20 20	JR	NZ, 4F9F
4F7F:	2A 20 40	LD	HL, (4020)
4F82:	11 40 00	LD	DE, 0040
4F85:	19	ADD	HL, DE
4F86:	7C	LD	A, H
4F87:	FE 40	CP	40
4F89:	20 14	JR	NZ, 4F9F
4F8B:	CD 49 00	CALL	0049
4F8E:	FE 01	CP	01
4F90:	28 11	JR	Z, 4FA3
4F92:	FE 5B	CP	5B
4F94:	28 09	JR	Z, 4F9F
4F96:	FE 1F	CP	1F
4F98:	20 F1	JR	NZ, 4F8B
4F9A:	CD C9 01	CALL	01C9
4F9D:	F1	POP	AF
4F9E:	C9	RET	
4F9F:	F1	POP	AF
4FA0:	03 58 04	JP	0458
4FA3:	3E 0E	LD	A, 0E
4FA5:	32 22 40	LD	(4022), A
4FA8:	C3 CC 06	JP	06CC
4FAB:	E5	REPEAT PUSH	HL
4FAC:	CD E0	CALL	4FE0
4FAF:	21 00 03	LD	HL, 0300
4FB2:	A7	AND	A
4FB3:	28 05	JR	Z, 4FBA
4FB5:	32 FE	LD	(4FFE), A
4FB8:	18 21	JR	4FDB
4FBA:	21 01 38	LD	HL, 3801
4FBD:	B6	OR	(HL)
4FBE:	CB 25	SLA	L
4FC0:	F2 BD	JP	P, 4FBD
4FC3:	A7	AND	A
4FC4:	28 15	JR	Z, 4FDB
4FC6:	2A FC	LD	HL, (4FFC)
4FC9:	2B	DEC	HL
4FCA:	7C	LD	A, H
4FCB:	A7	AND	A
4FCC:	3E 00	LD	A, 00
4FCE:	20 0B	JR	NZ, 4FDB
4FD0:	3E 03	LD	A, 03
4FD2:	32 FA	LD	(4FFA), A
4FD5:	21 A0 01	LD	HL, 01A0
4FD8:	3A FE	LD	A, (4FFE)
4FDB:	22 FC	LD	(4FFC), HL
4FDE:	E1	POP	HL
4DDF:	C9	RET	
4FE0:	E5	FLASH PUSH	HL
4FE1:	2A FA	LD	HL, (4FFA)
4FE4:	2B	DEC	HL
4FE5:	7C	LD	A, H
4FE6:	A7	AND	A
4FE7:	20 0A	JR	NZ, 4FF3
4FE9:	2A 20 40	LD	HL, (4020)
4FEC:	7E	LD	A, (HL)
4FED:	EE 00	OR	00
4FEF:	77	LD	(HL), A
4FF0:	21 00 03	LD	HL, 0300
4FF3:	22 FA	LD	(4FFA), HL
4FF6:	E1	POP	HL
4FF7:	C3 F3 03	JP	03E3
4FFA:	00	NOP	
4FFB:	03	INC	BC
4FFC:	00	NOP	
4FFD:	03	INC	BC
4FFE:	00	NOP	
4FFF:	00	NOP	

			<u>NAME</u>	util3
42E9:	F5		PUSH	AF
42EA:	21	49	LD	HL,
42EB:	79		LD	A,C
42EE:	E6	FE	AND	FE
42F0:	20	08	JR	NZ,42F5
42F2:	71		LD	(HL),C
42F3:	18	20	JR	4322
42F5:	7E		LD	A,(HL)
42F6:	B7		OR	A
42F7:	28	28	JR	Z,4324
42F9:	3A	22 40	LD	A,(4022)
42FC:	B7		OR	A
42FD:	20	25	JR	NZ,4324
42FF:	79		LD	A,C
4300:	FE	00	CP	00
4302:	20	20	JR	NZ,4324
4304:	2A	20 40	LD	HL,(4020)
4307:	11	40 00	LD	DE,0040
430A:	19		ADD	HL,DE
430B:	7C		LD	A,H
430D:	F1	40	CP	40
430E:	20	14	JR	NZ,4324
4310:	0D	49 00	CALL	0049
4313:	FE	01	CP	01
4317:	FE	5B	CP	5B
4319:	28	09	JR	Z,4324
431B:	FE	1F	CP	1F
431D:	20	F1	JR	NZ,4310
431F:	CD	C9 01	CLS	CALL 01C9
4322:	F1		BYPASS	POP AF
4323:	C9		RET	
4324:	F1		RST	POP AF
				JUMP TO LWRCSE
4327:	3E	0E	READY	LD A,0E
4329:	32	22 40	LD	(4022),A
432C:	F3	0C 04	JP	040C
432F:	DD	6E 03	LWRCSE	LD L,(IX-03)
4332:	DD	66 04	LD	H,(IX+04)
4335:	DA	9A 04	JP	C,049A
4338:	DD	7E 05	LD	A,(IX+05)
433B:	87		OR	A
433D:	28	01	JR	Z,433F
433E:	77		LD	(HL),A
433F:	79		LD	A,C
4340:	FE	20	CP	20
4342:	DA	04 05	JP	C,0504
4345:	FE	00	CP	00
4347:	30	08	JR	NC,4351
4349:	FE	80	CP	80
434D:	D2	A6 04	JP	NC,04A6
434E:	C3	7D 04	JP	047D
4351:	D6	00	SUB	00
4353:	18	F9	JR	434E
4355:	E5		REPEAT	PUSH HL
4359:	21	00 03	LD	HL,0300
435C:	A7		AND	A
435D:	28	05	JR	Z,4364
4362:	18	21	JR	4385
4364:	31	01 38	NOT NEW	LD HL,3801
4367:	B6		KLOOP	OR (HL)
4368:	CB	25	SLA	L
436D:	A7		AND	A
436E:	28	15	JR	Z,4385
4373:	2B		DEC	HL
4374:	7C		LD	A,H
4375:	A7		AND	A
4376:	3E	00	LD	A,00
4378:	20	0B	JR	NZ,4385
437A:	3E	03	LD	A,03
437C:	32	A4 43	LD	(43A4),A

```

437F: 21 A0 01      LD  HL,01A0

4388: E1            POP  HL
4389: 09            RET
438A: E5            FLASH PUSH HL

438E: 2B            DEC  HL
438F: 7C            LD   A,H
4390: A7            AND  A
4391: 20 0A          JR   NZ,439D
4393: 2A 20 40       LD   HL,(4020)
4396: 7E            LD   A,(HL)
4397: FE D0         XOR  D0
4399: 77            LD   (HL),A

```

```

43A0: E1            POP  HL
43A1: 03 E3 03       JP   03E3
43A4: 00            NOP
43A5: 03            INC  BC
43A6: 00            NOP
43A7: 03            INC  BC
43A8: 00            NOP
43A9: 00            NOP
43AA: 21 F9 42     INIT LD   HL,42E9
43AD: 22 1E 40     LD   (401E),HL

```

```

43B3: 22 16 40     LD   (4016),HL

```

CHANGE AA to CF TO PREVENT OVERWRITING
OF INIT.

```

43B9: AF            XOR  A
43BA: 77            LD   (HL),A
43BB: 23            INC  HL
43BC: 77            LD   (HL),A
43BD: 22 A4 40     LD   (40A4),HL
43C0: 23            INC  HL
43C1: 77            LD   (HL),A
43C2: 23            INC  HL
43C3: 22 F9 40     LD   (40F9),HL
43C6: 22 FB 40     LD   (40FB),HL
43C9: 22 FD 40     LD   (40FD),HL
43CC: C3 CC 06     JP   06CC

```

```

NAME      UTIL 4
4F30: 21 3F      INIT LD   HL,4F3F
4F33: 22 1E 40   LD   (401E),HL
4F36: 21 AB      LD   HL,4FAB
4F39: 22 16 40   LD   (4016),HL
4F3C: C3 CC 06   JP   06CC      RETURN TO BASIC
4F3F: F5      SCROLL PUSH AF
4F40: 21 FF      LD   HL,4FFF
4F43: 79          LD   A,C
4F44: E6 FE      AND  FE
4F46: 20 03      JR   NZ,4F4B
4F48: 71          LD   (HL),C
4F49: 18 2D      JR   417B
4F4B: 7E          LD   A,(HL)
4F4C: B7          OR   A
4F4D: 28 2B      JR   Z,4F7A
4F4F: 3A 22 40   LD   A,(4022)
4F52: B7          OR   A
4F53: 20 25      JR   NZ,4F7A
4F55: 79          LD   A,C
4F56: FE 0D      CP   0D
4F58: 20 20      JR   NZ,4F7A
4F5A: 2A 20 40   LD   HL,(4020)
4F5D: 11 40 00   LD   DE,0040
4F60: 19          ADD  HL,DE
4F61: 7C          LD   A,H
4F62: FE 40      CP   40
4F64: 20 14      JR   NZ,4F7A
4F66: CD 49 00   CALL 0049
4F69: FE 01      CP   01
4F6B: 28 10      JR   Z,4F7D

```

4F6D:	FE 5B	CP	5B
4F6F:	28 09	JR	Z, 4F7A
4F71:	FE 1F	CP	1F
4F73:	20 F1	JR	NZ, 4F66
4F75:	CD 09 01	CALL	01C9
4F78:	F1	POP	AF
4F79:	C9	RET	
4F7A:	F1	POP	AF
4F7B:	18 08	JR	4F85
4F7D:	3E 0E	LD	A, 0E
4F7F:	32 22 40	LD	(4022), A
4F82:	03 0C 06	JP	060C
4F85:	DD 6E 03	LWRCSE	LD L, (1X+03)
4F88:	DD 66 04		LD H, (1X+04)
4F8B:	DA 9A 04	JP	C, 049A
4F8F:	DD 7E 05	LD	A, (1X+05)
4F91:	B7	OR	A
4F92:	28 01	JR	Z, 4F95
4F94:	77	LD	(HL), A
4F95:	79	LD	A, C
4F96:	FE 20	CP	20
4F98:	0A 06 05	JP	C, 0506
4F9B:	FE 10	CP	10
4F9D:	30 08	JR	NC, 4FA7
4F9F:	FE 80	CP	80
4FA1:	D2 A6 04	JP	NC, 04A6
4FA4:	C3 7D 04	JP	047D
4FA7:	D6 00	SUB	C0
4FA9:	18 F9	JR	4FA4
4FAB:	F5	REPEAT	PUSH HL
4FAC:	CD 10		CALL 4FE0
4FAF:	21 00 03		LD HL, 0300
4FB2:	A7		AND A
4FB3:	28 05	JR	Z, 4FBA
4FB5:	32 FE	LD	(4FFE), A
4FB8:	18 21	JR	4FDB
4FBA:	21 01 38	LD	HL, 3801
4FBD:	B6	OR	(HL)
4FBE:	CB 25	SLA	L
4FCC:	F2 B3	JP	P, 4FBD
4FC3:	A7	AND	A
4FC4:	28 15	JR	Z, 4FDB
4FC6:	2A FC	LD	HL, (4FFC)
4FC9:	2B	DEC	HL
4FCA:	7C	LD	A, H
4FCB:	A7	AND	A
4FCC:	3E 00	LD	A, 00
4FCE:	20 08	JR	NZ, 4F0B
4FDO:	3E 03	LD	A, 03
4FD2:	32 FA	LD	(4FFA), A
4FD5:	21 A0 01	LD	HL, 01A0
4FDB:	3A FE	LD	A, (4FFE)
4FDB:	22 FC	LD	(4FFC), HL
4FDE:	E1	POP	HL
4FDF:	09	RET	
4FE0:	E5	FLASH	PUSH HL
4FE1:	2A FA		LD HL, (4FFA)
4FE4:	2B	DEC	HL
4FE5:	7C	LD	A, H
4FE6:	A7	AND	A
4FE7:	20 0A	JR	NZ, 4FF3
4FE9:	2A 20 40	LD	HL, (4020)
4FEC:	7E	LD	A, (HL)
4FED:	EE D0	XOR	D0
4F1F:	77	LD	(HL), A
4FF0:	21 00 03	LD	HL, 0300
4FF3:	22 FA	LD	(4FFA), HL
4FF6:	E1	POP	HL
4FF7:	C3 E3 03	JP	03E3
4FFA:	00	NOP	
4FFB:	03	INC	BC
4FFC:	00	NOP	
4FFD:	03	INC	BC
4FFE:	00	NOP	
4FFF:	00	NOP	

***** NEXT MONTH'S ISSUE *****

Next month's issue will contain at least the following programs plus the usual features and articles.

** ROVING TARGETS LI/4K **

Another educational program that teaches young children equivalence in fractions by combining learning with a game. While they play, they learn.

** SOLITAIRE LII/16K **

This old favourite at last comes to your '80 complete with graphics. There is only one thing wrong with playing solitaire on your computer - you can't cheat !!!

** SEMAPHORE LII/4K **

This program is a fun way for boy scouts or anybody else for that matter, to learn the semaphore system of signalling with hand flags.

** MOVIE m/1 **

This machine language program adds two more commands to the BASIC interpreter which are useful in animated displays as the commands let you move part or all of the screen display left or right, with or without wrap around, at machine language speed.

** THREE BILLY GOATS GRUFF LII/16K **

This program is just the thing to keep the kids quiet for a while - a story complete with pictures of the well-known children's tale. Watch out for the TROLL !!!

APPLICATION FOR PUBLICATION
OF A PROGRAM
IN MICRO-80

To MICRO-80
Please consider the enclosed program for . . .

Date

Tick where appropriate

(i) Publication in MICRO-80

(ii) Publication on disk or cassette only

(iii) Both

Name

Address

.

Postcode

*** CHECK LIST ***

- Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.
- Ensure that you supply adequate instructions, notes on what the program does and how it does it, etc.
- For system tapes, the start, end, and entry points, etc.
- The changes or improvements that you think may improve it.
- Please package securely — padabags are suggested — and enclose stamps or postage if you want your cassette or disk returned.

***** CASSETTE EDITION INDEX *****

The cassette edition of MICRO-80 contains all the software listed each month, on cassette. All cassette subscribers need do is CLOAD and RUN the programs. Level II programs are recorded on side 1 of the cassette. Level I programs are recorded on side 2. Level I programs are not compatible with the System 80. All programs are recorded twice in succession. Note, System 80 computers have had different tape-counters fitted at different times. The approximate start positions shown are correct for the very early System 80 without the volume control or level meter. They are probably incorrect for later machines. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.

SIDE ONE	TYPE	I.D.	APPROX. START POSITION		
			CTR-41	CTR-80	SYSTEM-80
BINARY CONVERTER	LII/4K	B	10	7	7
"	"	"	19	13	14
LINEAR REGRESSION	LI/4K	L	30	20	21
"	"	"	52	35	37
BIG LETTERS	LII/16K	A	74	50	52
"	"	"	170	115	120
UTIL 1	m/1	UTIL 1	255	173	182
"	"	"	260	176	185
UTIL 2	m/1	UTIL 2	264	179	188
"	"	"	269	182	191
UTIL 3	m/1	UTIL 3	273	185	194
"	"	"	278	188	197
UTIL 4	m/1	UTIL 4	282	191	200
"	"	"	286	194	204
COSMIC WARS	32K/DISK	C	290	197	207
"	"	"	310	210	221
SIDE TWO					
BIRTHDAY	LI/4K	-	10	7	-
"	"	-	74	50	-
GUZINTA	LI/4K	-	133	90	-
"	"	-	188	127	-

TO:
MICRO-80, P.O. BOX 213, GOODWOOD,
SOUTH AUSTRALIA. 5034.

Please RUSH to me the items shown below:

\$ enclosed Date

..... 12 month subscription to MICRO-80

..... 12 month subscription to MICRO-80, plus the
cassette edition

..... The latest issue of MICRO-80
(see inside front cover for prices)

The MICRO-80 PRODUCTS listed below:

DESCRIPTION	PRICE

TOTAL ENCLOSED WITH ORDER

☐ Cheque ☐ Bankcard ☐ Money Order

Bankcard Account Number


--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Signature Exp. End

NAME

ADDRESS

Postcode



BANKCARD



A SPECIAL OFFER!

**TO
SUBSCRIBERS OF**

MICRO 80

FOR THE FIRST TIME IN AUSTRALIA!
GENUINE TANDY TRS-80 MICRO-COMPUTERS,
HARDWARE AND SOFTWARE AT

GREAT DISCOUNT PRICES!!!

HOW TO ORDER -

1. SELECT ITEMS FROM 1980 TANDY CATALOGUE
2. DEDUCT 10% FROM ADVERTISED PRICES
3. POST US YOUR ORDER STATING DESCRIPTION
CAT. No AND A CHEQUE OR MONEY ORDER.

WE WILL -

1. ATTEND TO YOUR ORDER WITHIN 7 DAYS
2. SUPPLY GOODS SELECTED FREIGHT FREE !
3. SEND ADVERTISING REGULARLY TO KEEP YOU INFORMED OF CURRENT SPECIALS !

* subject to availability



CONQUEST ELECTRONICS

PTY. LTD.

212 KATOOMBA ST KATOOMBA N.S.W. 2780 PHONE (047) 82 2491

[illegible]

MICRO-80

LEVEL II ROM REFERENCE MANUAL

by Edwin Paay

Published by MICRO-80 PRODUCTS

Written by Eddy Paay, the LEVEL II ROM REFERENCE MANUAL is the most complete explanation of the Level II BASIC interpreter ever published.

Part 1 lists all the useful and usable ROM routines, describes their functions explains how to use them in your own machine language programs and notes the effect of each on the various Z 80 registers.

Part 1 also details the contents of system RAM and shows you how to intercept BASIC routines as they pass through system RAM. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory—the only restriction is your own imagination!

Part 2 gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements, etc. It also describes the various formats used for BASIC, SYSTEM and EDITOR/ASSEMBLER tapes. Each section is illustrated by sample programs which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

The LEVEL II ROM REFERENCE MANUAL is intended to be used by machine language programmers. It assumes a basic understanding of the Z 80 instruction set and some experience of Assembly Language programming. But BASIC programmers too will benefit from reading it. They will gain a much better insight into the functioning of the interpreter which should help them to write faster, more concise BASIC programs.

MICRO-80